ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΡΜΗΝΕΥΣΙΜΑ ΜΟΝΤΕΛΑ ΒΑΘΙΑΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΗΝ ΑΝΑΓΝΩΡΙΣΗ ΕΙΚΟΝΑΣ: ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΤΕΧΝΙΚΩΝ ΕΠΕΞΗΓΗΣΙΜΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Ηλίας Ευθυμίου

Α.Μ. 713242017010

Επιβλέπων καθηγητής: Επ. Καθηγητής Χρήστος Τρούσσας

Ακαδημαϊκό έτος 2023-2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΡΜΗΝΕΥΣΙΜΑ ΜΟΝΤΕΛΑ ΒΑΘΙΑΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΗΝ ΑΝΑΓΝΩΡΙΣΗ ΕΙΚΟΝΑΣ: ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΤΕΧΝΙΚΩΝ ΕΠΕΞΗΓΗΣΙΜΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Ηλίας Ευθυμίου
Α.Μ. 713242017010

Επιβλέπων καθηγητής:

Επ. Καθηγητής Χρήστος Τρούσσας

Εξεταστική Επιτροπή:

Χρήστος Τρούσσας (Επ. Καθηγητής)	Ακριβή Κρούσκα (Μέλος ΕΔΙΠ)	Παναγιώτα Τσελέντη (Μέλος ΕΔΙΠ)
Christos Troussas Digitally signed by Christos Troussas Date: 2024.03.20 10:59:57 +02'00'	Akrivi Krouska Digitally signed by Akrivi Krouska Date: 2024.03.21 11:30:22 +02'00'	Panagiota Tselenti Digitally signed by Panagiota Tselenti Date: 2024.03.20 20:28:33 +02'00'

Ημερομηνία εξέτασης __/3/2024

Δήλωση Συγγραφέα Διπλωματικής Εργασίας

Ο κάτωθι υπογεγραμμένος Ευθυμίου Ηλίας του Κωνσταντίνου, με αριθμό μητρώου 713242017010 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Ηλίας Ευθυμίου

Ευχαριστίες

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο. Επίσης θέλω να ευχαριστήσω θερμά τον καθηγητή Τρούσσα Χρήστο για την υπομονή και τον χρόνο που διέθεσε.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά και τους φίλους μου, για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την εξερεύνηση τεχνικών επεξηγήσιμης τεχνητής νοημοσύνης για μοντέλα βαθιάς μάθησης που χρησιμοποιούνται στην αναγνώριση εικόνων. Πιο συγκεκριμένα, θα συγκριθούν και θα αξιολογηθούν διαφορετικές προσεγγίσεις για την εξήγηση της διαδικασίας λήψης αποφάσεων αυτών των μοντέλων, συμπεριλαμβανομένων μεθόδων που βασίζονται σε απόδοση, saliency maps και τεχνικών που βασίζονται σε κλίση (gradient-based techniques). Καθώς θα διερευνηθεί πως αυτές οι τεχνικές μπορούν να βοηθήσουν τους χρήστες να κατανοήσουν και να εμπιστευτούν τις αποφάσεις που λαμβάνονται απο αυτα τα μοντέλα. Η έρευνα περιλαμβάνει εκτός από αναλυτικό θεωρητικό υπόβαθρο και πειραματισμό πάνω σε μοντέλα βαθιάς μάθησης με όσο καλύτερα δεδομένα μπορούμε να έχουμε. Το σύνολο δεδομένων το οποίο θα χρησιμοποιηθεί στο πείραμα είναι το MNIST, ένα ανοικτό σύνολο δεδομένων και αρκετά γνωστό για εκπαίδευση μοντέλων αναγνώρισης εικόνας. Κατά τον πειραματισμό έχει γίνει χρήση της γλώσσας προγραμματισμού python καθώς είναι ένα αρκετά εύκολο και δυνατό εργαλείο που χρησιμοποιείται αρκετά τα τελευταία χρόνια από την επιστημονική κοινότητα, ειδικά στα πλαίσια της τεχνητής νοημοσύνης. Όσο για την εκπαίδευση των μοντέλων μηχανικής μάθησης που θα χρησιμοποιηθούν εχει επιλεγεί το framework Pytorch καθώς είναι open-source και python-friendly εργαλείο.

Λέξεις κλειδιά

Explainable Artificial Intelligence, Αναγνώριση Εικόνας, Pytorch, LIME, DeepSHAP

Abstract

This thesis deals with the exploration of explainable artificial intelligence techniques for deep learning models used in image recognition. More specifically, different approaches to explain the decision-making process of these models will be compared and evaluated, including performance-based methods, saliency maps and gradient-based techniques. As it will be explored how these techniques can help users understand and trust the decisions made by these models. The research includes analytical theoretical background and experimentation on deep learning models with the best data we can get. The dataset that will be used in the experiment is MNIST, an open dataset and quite well known for training image recognition models. During the experimentation, the python programming language has been used as it is a fairly easy and powerful tool that has been used quite a lot in recent years by the scientific community, especially in the context of artificial intelligence. As for the training of the machine learning models that will be used, the Pytorch framework has been chosen as it is an open-source and python-friendly tool.

Key words

Explainable Artificial Intelligence, Image Recognition, Pytorch, LIME, DeepSHAP

Κατάλογος σχημάτων και εικόνων

Περιεχόμενα

1 Εισαγωγή

Τα τελευταία χρόνια με την άνοδο της τεχνητής νοημοσύνης και ειδικότερα της μηχανικής μάθησης η ανθρωπότητα έχει δεί διάφορα επιτεύγματα της, επηρεάζοντας αποφάσεις των χρηστών που το χρησιμοποιούν, απο τις πιο απλές αποφάσεις (έξυπνα συστήματα που μπορεί προτείνουν τουριστικούς προορισμούς) μέχρι και τις πιο σημαντικές (ένταξη έξυπνων συστημάτων στην αυτόματη οδήγηση).

Όμως παρά το αποτέλεσμα που δίνουν αυτα τα συστήματα, είτε σωστά είτε εσφαλμένα, δεν δίνουν κάποιου είδους πληροφορία στον χρήστη που το χρησιμοποιεί ώστε να καταλάβει γιατί πάρθηκε αυτή η απόφαση και ούτε φαίνεται προς τα έξω η διαδικασία λήψης αποφάσεων. Λόγω αυτών των προβλημάτων υπάρχει απώλεια εμπιστοσύνης από τους χρήστες αλλα και απο τους μηχανικούς που τα αναπτύσουν. Έτσι για να λυθούν τέτοιου είδους προβλήματα δημιουργήθηκε ο κλάδος της επεξηγήσιμης τεχνητής νοημοσύνης που προσπαθεί να δώσει μεθόδους και τεχνικές ώστε να εδραιώσει την εμπιστοσύνη προς την τεχνητή νοημοσύνη από τους χρήστες.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η εξερεύνηση τεχνικών επεξηγήσιμης τεχνητής νοημοσύνης για μοντέλα βαθιάς μάθησης που χρησιμοποιούνται στην αναγνώριση εικόνων. Με στόχο να συγκριθούν και να αξιολογηθούν διαφορετικές προσεγγίσεις για την εξήγηση της διαδικασίας λήψης αποφάσεων αυτών των μοντέλων, συμπεριλαμβανομένων μεθόδων που βασίζονται σε απόδοση, saliency maps και τεχνικων που βασίζονται σε κλίση (gradient-based techniques). θα διερευνηθεί πως αυτές οι τεχνικές μπορούν να βελτιώσουν την ερμηνευσιμότητα και την διαφάνεια των μοντέλων βαθιάς μάθησης και πως μπορουν να βοηθήσουν τους χρήστες να κατανοήσουν και να εμπιστευτούν τις αποφάσεις που λαμβάνονται απο αυτα τα μοντέλα. Καθώς και πειραματισμό με τις διάφορες μεθόδους.

Στα κεφάλαια που θα ακολουθήσουν, θα αναπτυχθεί το θεωρητικό υπόβαθρο της εργασίας ώστε να κατανοήσουμε τον ρόλο της επεξηγήσιμης τεχνητής νοημοσύνης, τις μεθόδους και τεχνικές που θα χρησιμοποιήσουμε στο πρακτικό κομμάτι αλλά και να απαντήσουμε στα ερωτήματα που θέσαμε. Για το πρακτικό κομμάτι θα γίνει χρήση της γλώσσας προγραμματισμού python η οποία είναι αρκετα συνηθισμένη στον χώρο της τεχνητής νοημοσύνης τα τελευταία χρόνια, με ποικίλες βιβλιοθήκες, όπως και frameworks που βοηθούν στην δημιουργία μοντέλων βαθιάς μάθησης. Το framework που θα χρησιμοποιηθεί για την εξαγωγή ενός τέτοιου μοντέλου είναι το Pytorch. Ένα framework το οποίο είναι python-friendly και αρκετά σύνηθες τα τελευταία χρόνια για ερευνητικό σκοπό.

Πιο συγκεκριμένα η παρούσα εργασία είναι οργανωμένη ως εξής: Το πρώτο κεφάλαιο αποτελείται από μια εισαγωγή για τον στόχο της εργασίας, τα ερωτήματα που έχει να θέσει καθώς και μια μικρή αναφορά για τα εργαλεία που θα χρησιμοποιηθούν στην πορεία της εργασίας. Στο δεύτερο κεφάλαιο αναλύεται το θεωρητικό υπόβαθρό που είναι αναγκαίο για την καλύτερη κατανόηση της εργασίας, πιο συγκεκριμένα αναλύει τι είναι επεξηγήσιμη τεχνητή νοημοσύνη, τις μεθόδους και τεχνικές που χρησιμοποιούνται ιδιαίτερα πάνω σε μοντέλα βαθιάς μάθησης για την αναγνώριση εικόνας. Στο τρίτο και τέταρτο κεφάλαιο περιγράφεται αναλυτικά το πειραματικό μέρος, τι εργαλεία χρησιμοποιούνται, πιθανά προβλήματα και σχολιασμό των αποτελεσμάτων. Τέλος παρουσιάζονται τα συμπεράσματα μας απο την διεξαγωγή του πειράματος μας.

2 Θεωρητικό υπόβαθρο

Αυτό το κεφάλαιο έχει ως σκοπό να βοηθήσει τον αναγνώστη να κατανοήσει τις βασικές αρχές της επεξηγήσιμης τεχνητής νοημοσύνης και τον ρόλο της στην σύγχρονη εποχή καθώς και να μπορέσει να κάνει μια πρώτη επισκόπηση στις διάφορες μεθόδους και τεχνικές που έχουν προταθεί ανα περιόδους στον κλάδο.

2.1 Τι είναι επεξηγήσιμη τεχνητή νοημοσύνη

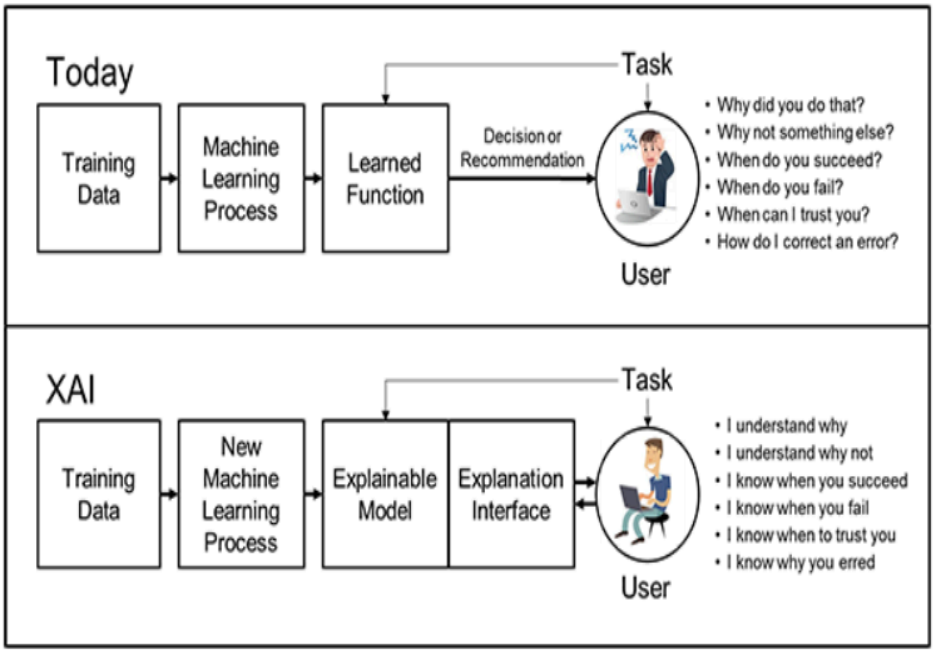
Η επεξηγήσιμη τεχνητή νοημοσύνη (XAI) είναι ένας κλάδος με αρκετά μεγάλη κοινότητα που καταπιάνεται με αυτόν εδώ και αρκετά χρόνια. Σκοπός αυτού του κλάδου είναι η δημιουργία τεχνικών και μεθόδων που θα βοηθήσουν τον άνθρωπο να κατανοήσει και να εξηγήσει με καλύτερο τρόπο τις αποφάσεις των μοντέλων μηχανικής μάθησης όπως είναι των DNNs (deep neural networks), να δώσει τέτοιες εξηγήσεις ώστε να αποκαλύψει σε έναν βαθμό τις λογικές που κρύβονται κάτω από το μοντέλο[1] και να βοηθήσει να εδραιωθεί η εμπιστοσύνη του ανθρώπου προς συστήματα τεχνητής νοημοσύνης. Η εμπιστοσύνη είναι θέμα που φαίνεται να έχει απασχολήσει την κοινότητα αρκετά καθώς τέτοια συστήματα έχουν εισαχθεί και σε κρίσιμες εφαρμογές, οπότε μια τέτοια συζήτηση σχεδόν ποτέ δεν λείπει. Ωστόσο ενώ φαίνεται για αρχή αυτά να είναι τα πιο κρίσιμα χαρακτηριστικά με τα οποία ασχολείται η κοινότητα, ανα περιόδους έχει εισάγει διάφορες προκλήσεις που θα κληθεί να δώσει λύση. Σαν κλάδος εχει δει αρκετά σημαντική πρόοδο και έχει δημιουργήσει αρκετά μεγάλη ποικιλία από μεθόδους και τεχνικές που προσπαθούν να βοηθήσουν στην καλύτερη κατανόηση των αποτελεσμάτων ενός συστήματος τεχνητής νοημοσύνης αλλά και να αναδείξουν τις δυνατότητες και αδυναμίες ενός συστήματος στον χρήστη[4].

Γενικά ο κλάδος της επεξηγήσιμης τεχνητής νοημοσύνης δεν είναι κάτι νέο, είχε εμφανιστεί και παλαιότερα, στα μέσα του 1980, με σκοπό να εξηγήσει και να κατανοήσει συστήματα βασισμένα σε κανόνες (rule-based expert systems)[1][3]. Αλλά με την σύγχρονη άνοδο των νεων μοντέλων μηχανικής μάθησης που χρησιμοποιούν μοντέλα βαθιάς μάθησης αναδεικνύεται ξανά η αναγκαιότητα του κλάδου, καθώς τέτοια μοντέλα είναι ιδιαίτερα αδιαφανή και έχουν αρκετή πολυπλοκότητα δημιουργώντας καταστάσεις και αποτελέσματα που δεν μπορούν να εξηγηθούν εύκολα αφου αυτά τα συστήματα δεν βγάζουν προς τα έξω κάποια τεκμηρίωση για τα αποτελέσματα τους.

Ενώ υπάρχει, εδώ και αρκετό καιρο φαίνεται ότι δεν έχει δοθεί κάποιος συγκεκριμένος αυστηρός ορισμός από την κοινότητα που θα εξηγεί τον ρόλο του κλάδου με σαφήνεια. Παρά τις προσπάθειες αρκετών, οι ορισμοί που έχουν δοθεί είναι συχνά ελλιπείς καθώς ο ορισμός μπορεί να μην καλύπτει διάφορες προκλήσεις που έχουν εισαχθεί από την κοινότητα[2]. Δεν ξεχνάμε ότι είναι ένας κλάδος που αναπτύσσεται γρήγορα και πολλά πράγματα μπορεί να είναι δύσκολο να εισαχθούν σε έναν ορισμό καθώς έχει αυξηθεί και η πολυπλοκότητα του ίδιου του κλάδου. Ένας από τους ορισμούς που είχε δοθεί για την επεξηγήσιμη τεχνητή νοημοσύνη είχε δοθεί από τον D. Gunning [2]: “XAI θα δημιουργήσει μια σουίτα από τεχνικές μηχανικής μάθησης που θα βοηθήσουν τους χρήστες να κατανοήσουν, εμπιστευτούν και να διαχειριστούν αποτελεσματικά την αναδυόμενη γενιά τεχνητής νοημοσύνης”. Είναι ένας ορισμός που είναι κατανοητός και αναφέρει όπως είπαμε την εμπιστοσύνη και την κατανόηση όμως αποτυγχάνει να συμπεριλάβει άλλες προοπτικές του κλάδου όπως ερμηνεύσιμα μοντέλα τεχνητής νοημοσύνης, αιτιότητα, δυνατότητα μεταφοράς, πληροφόρηση, και δικαιοσύνη.

Ωστόσο λόγω της αυξημένης ζήτησης του τομέα τον Αύγουστο του 2020 ο NIST δημοσίευσε τις τέσσερις βασικές αρχές τις οποίες πρέπει να ακολουθεί ένα σύστημα τεχνητής νοημοσύνης ώστε να μπορεί να θεωρηθεί ώς επεξηγήσιμη τεχνητή νοημοσύνη[3][14]. Η αρχή της εξήγησης

(explanation) αναφέρει ότι ένα σύστημα τεχνητής νοημοσύνης πρέπει να μπορεί να δώσει στοιχεία και να μπορεί να υποστηρίξει με αιτιολόγηση κάθε απόφαση που έχει παρθεί απο το συστημα. Η αρχή της χρήσιμης αιτιολόγησης (meaningful) αναφέρει ότι μια αιτιολόγηση που θα παρέχεται από ένα σύστημα τεχνητής νοημοσύνης θα πρέπει να είναι κατανοητή και με νόημα για τους χρήστες που θα παρέχεται. Καθώς ένα σύστημα θα μπορούσε να έχει διαφορετικές ομάδες χρηστών που αναφέρεται έτσι και η αιτιολόγηση θα πρέπει να βοηθάει ανάλογα. Η αρχή της ακρίβειας (accuracy) αναφέρει ότι η αιτιολόγηση που παρέχεται από το σύστημα τεχνητής νοημοσύνης θα πρέπει να ανακλά με ακρίβεια τις λειτουργίες του συστήματος. Και τέλος η αρχή για όρια γνώσης (knowledge limits), αναφέρει ότι τα συστήματα τεχνητής νοημοσύνης θα πρέπει να μπορούν να αναγνωρίζουν περιπτώσεις όπου το σύστημα δεν φτιάχτηκε για να της απαντήσει, καθώς δεν θα είναι συμβατές.



Εικόνα 1. Τι είναι επεξηγήσιμη τεχνητή νοημοσύνη απο το DARPA[33]

2.2 Γιατί χρειαζόμαστε την επεξηγήσιμη τεχνητή νοημοσύνη

Ενώ τα πρώτα συστήματα τεχνητης νοημοσύνης ήταν πολύ εύκολα στην κατανόηση, στην σύγχρονη εποχή έχουμε μια μεγάλη αύξηση των μοντέλων μηχανικής μάθησης που είναι αδιαφανείς και συχνά αποκαλούνται από την κοινότητα μαυρα κουτια (black-box). Τετοια μοντέλα λαμβάνουν διάφορες αποφάσεις, απο τις πιο απλές αποφάσεις μέχρι και τις πιο σημαντικές, επηρεάζοντας την ζωή πολλών ανθρώπων σε καθημερινή βάση. Συνήθως είναι αλγόριθμοι μηχανικής μάθησης, δηλαδή δεν έχουν γραφτεί από κάποιον άνθρωπο αλλά είναι εκπαιδευμένα από διάφορες συλλογές δεδομένων. Οπότε τα δεδομένα είναι αυτά που θα επηρεάσουν αργότερα την συμπεριφορά του συστήματος ως έναν βαθμό καθώς από αυτά εκπαιδευεται. Αυτό μπορεί να δημιουργήσει διάφορα θέματα στις αποφάσεις του συστήματος καθώς εάν υπάρχει κάποιου είδους προκαταλήψεις (bias) στα δεδομένα, τότε και το ίδιο το σύστημα μπορεί να εμφανίσει κάποιου είδους προκαταλήψεις. Άλλο ένα σημαντικό πρόβλημα που έχει εμφανιστεί από αυτά τα συστήματα, όπως είναι τα νευρωνικά δίκτυα, είναι η αδιαφάνεια αυτων των συστημάτων, δηλαδή αυτα τα συστήματα δεν μας βοηθούν με κάποιο τρόπο να αποκτίσουμε πληροφορίες για τον τρόπο με τον οποίο πήραν κάποια απόφαση.

Η επεξηγήσιμη τεχνητή νοημοσύνη με σκοπό να μην περιορίσει τα συστήματα τεχνητής νοημοσύνης τελευταίας γενιάς προσπαθεί να δημιουργήσει μια σουίτα από τεχνικές ώστε να

βοηθήσει του ανθρώπους να κατανοήσουν τον λόγο και τον τρόπο με τον οποίο κάποιο σύστημα τεχνητής νοημοσύνης έλαβε μια απόφαση καθώς και να δημιουργηθούν μοντέλα τεχνητής νοημοσύνης που να είναι διαφανεί όμως παράλληλα να μπορέσουμε να επωφεληθούμε από τις δυνατότητες τους όπως είναι η ακρίβεια στις προβλέψεις.

Ωστόσο πρέπει να αναφέρουμε ότι τα ευφυή συστήματα κατέχουν πολύ μεγάλη δύναμη και είναι αναγκαίο να υπάρχει ο τρόπος να το ελέγξουμε και να μπορούμε να εξηγήσουμε τον τρόπο με τον οποίο ένα τέτοιο σύστημα λαμβάνει τις αποφάσεις. Εάν μπορούμε να εξάγουμε τον τρόπο με τον οποίο λαμβάνει τις αποφάσεις του με έναν φιλικό τρόπο, τότε ένας χρήστης που αλληλεπιδρά μαζί του θα μπορεί να το εμπιστευτεί περισσότερο, καθώς οι άνθρωποι δεν εμπιστεύονται ένα σύστημα μονο λόγω των στατιστικών, αλλά και μέσα από τις αλληλεπιδρασεις του με αυτό. Εκτός από τους χρήστες τέτοιες μέθοδοι θα φανούν ιδιαίτερα χρήσιμοι και στους μηχανικούς ή ακόμα και ερευνητές ώστε να μπορέσουν να φτιαξουν καλυτερα συστήματα ή ακόμα και να αναβαθμίσουν τα είδη υπάρχοντα.

Η επεξηγησιμότητα τέτοιον συστημάτων είναι ιδιαίτερα κρίσιμη για τις εφαρμογές που θεωρούνται ιδιαίτερα κρίσιμες όπως υγεία και οικονομικούς εμπειρογνώμονες καθώς και συστήματα αυτόνομης οδήγησης. Ενα ευφυή σύστημα θα μπορούσε να επηρεάσει το πλάνο της θεραπείας ενός ασθενή ή ακόμα και να εμπλακεί σε κάποιο τροχαίο ατύχημα. Έτσι θα ήταν σημαντικό να γνωρίζουμε πότε ένα τέτοιο σύστημα μπορεί να αποτύχει ή ακόμα και να ξέρουμε γιατί απέτυχε όταν γίνει ένα συμβάν.

Επίσης φαίνεται ότι καθώς αυξάνονται οι εφαρμογές που χρησιμοποιούν μεθόδους τεχνητής νοημοσύνης και πιο συγκεκριμένα μηχανικής μάθησης, αυξάνεται και η ανάγκη να μπορεί να συνεργάζεται με τους νόμους [46]. Εισάγωντας νόμους σε τέτοια συστήματα η επεξηγησημότητα και η ανάγκη για κατανόηση των αποτελεσμάτων σε διαφορετικές ομάδες ανθρώπων θα είναι πιο κρίσιμη απο ποτε. Θα βοηθήσει κιόλας σε τέτοιες καταστάσεις και τους μηχανικούς για τον καλύτερο έλεγχο των συστημάτων και να δώσουν την καλύτερη δυνατή αιτιολόγηση εφόσον κάθε μέθοδος για αιτιολόγηση που θα φτιάχνεται από τον κλάδο θα μπορεί να δίνει και μια διαφορετική απεικόνιση της αιτιολόγησης του συστήματος σε διαφορετικά στάδια ανάπτυξης και χρήσης της ίδιας της εφαρμογής.

Επομένως για τις διάφορες εφαρμογές που θα συνδυάζονται αργότερα από κρισιμοτητα και την προστασία του ανθρώπου μέσω των διαφορετικών νόμων που μπορεί να επιβάλλονται κρίνεται αναγκαία και η δυνατότητα τέτοιων συστημάτων να μπορούν να παρέχουν διαφορετικά επίπεδα διαφάνειας [28]. Τα διαφορετικά επίπεδα θα παρέχουν προς το κοινο τις κατάλληλες λειτουργίες που μπορεί και υλοποιεί το σύστημα για να μπορέσουμε να λύσουμε κάποιο πρόβλημα.

Έτσι λόγω όλων των παραπάνω κρίνεται αναγκαίος ο κλάδος της επεξηγήσιμης τεχνητής νοημοσύνης, ώστε να μπορέσουμε να έχουμε μεθόδους καλύτερης κατανόησης αδιαφανων ευφυη συστημάτων αλλά και να μπορέσει να εδραιωθεί η εμπιστοσύνη απο τους χρήστες που θα αλληλεπιδράσουν μαζί τους. Παράλληλα με την αύξηση αυτού του κλάδου, υφιστάμενες μέθοδοι μηχανικής μάθησης που είναι δύσκολες στην κατανόηση θα γίνουν πιο κατανοητές απο τους ερευνητές και δεν θα τις εγκαταλείψουν.

2.3 Βασικοί ορισμοί

Σε αυτήν την ενότητα γίνεται αναφορά των βασικών ορισμών ώστε να μπορέσουμε να κατανοήσουμε με μεγαλύτερη ευκολία την γλώσσα που χρησιμοποιείται στην επεξηγήσιμη τεχνητή νοημοσύνη. Οι ορισμοί που θα συμπεριληφθούν είναι οι εξής: τι είναι ένα μαύρο κουτί, τι είναι ένα ερμηνευτής, τί είναι ένας ταξινομητής, ποιοί είναι οι ταξινομητές εντός τομέα και ποιοί μεταξύ τομέων, τι είναι μια εξήγηση, ποιά είναι τα εγγενώς ερμηνεύσιμα μοντέλα και ποια τα αδιαφανή μοντέλα, τι είναι η πιστότητα, ποιές είναι οι τοπικές εξηγήσεις και ποιές είναι οι παγκόσμιες, τι

εννοούμε όταν μιλάμε για posthoc εξηγήσεις και τι για antehoc, ποια είναι τα αντιπαραστατικά και τι οι αντιπαραστατικές εξηγήσεις, ποιές είναι οι διαβουλευτικές εξηγήσεις, ποιές είναι οι οπτικές εξηγήσεις και ποιές οι κειμενικές εξηγήσεις και τέλος τι είναι ερμηνευσιμότητα, η επεξηγησιμότητα και τη η διαφάνεια.

Τα μαύρα κουτιά είναι μοντέλα τα οποία είναι ιδιαίτερα αδιαφανή και χρειάζονται τεχνικές εξηγήσεων για να κατανοήσουμε το πως δουλεύουν. Στο πλαίσιο της παρούσας εργασίας τα μαύρα κουτιά θα θεωρούμε ότι είναι τύπου συνελικτικών νευρωνικών δικτύων (Convolutional Neural Networks) που συνήθως χρησιμοποιούνται σε εφαρμογές που κάνουν ταξινόμηση εικόνων.

Ένας ερμηνευτής δεν είναι τίποτα παραπάνω παρά μια αλγοριθμική διαδικασία που εξηγεί τους μηχανισμούς ενός μαύρου κουτιού. Ενώ ένας ταξινομητής είναι το μοντέλο που κάνει την ταξινόμηση μιας εισόδου σε μια προκαθορισμένη κατηγορία ή διαφορετικά κλάση.

Οι ταξινομητές εντός τομέα είναι ταξινομητές που έχουν εκπαιδευτεί και δοκιμαστεί πάνω σε δείγματα από την ίδια κατανομή ενώ στην αντίθετη περίπτωση οι ταξινομητές μεταξύ τομέων είναι ταξινομητές που έχουν εκπαιδευτεί και δοκιμαστεί σε δείγματα από διαφορετικές κατανομές.

Μια εξήγηση συχνά αναφέρεται ώς μια απλοποιημένη απεικόνιση του μηχανισμού λειτουργίας που χρησιμοποιεί το μαύρο κουτί, παρόλα αυτά ακόμη δεν έχει τεθεί ένας αυστηρός ορισμός που να συμπεριλαμβάνει την μορφή της εξήγησης.

Ο τρόπος με τον οποίο γίνονται οι εξηγήσεις στον κλάδο εξαρτάται πολύ από την φύση των μοντέλων μηχανικής μάθησης, δηλαδή εάν είναι εγγενώς ερμηνεύσιμα ή αδιαφανή μοντέλα. Στα εγγενώς ερμηνεύσιμα μοντέλα ανήκουν τα δέντρα αποφάσεων, Bayesian classifiers και sparse linear models, αυτά τα μοντέλα είναι πιο εύκολο να κατανοηθούν και να ερμηνευτούν οι αποφάσεις τους. Αυτοί οι αλγόριθμοι έχουν περιορισμένο αριθμό εσωτερικών στοιχείων αλλά προσφέρουν ιχνηλασιμότητα και διαφάνεια στην διαδικασία της λήψης αποφάσεων. Από την άλλη πλευρά στα αδιαφανή μοντέλα ανήκουν οι αλγόριθμοι βαθιάς μάθησης οι οποιοι θυσιάζουν την διαφάνεια ώστε να πετύχουν καλύτερη ακρίβεια στα αποτελέσματα τους. Τέτοιοι αλγόριθμοι έχουν χρησιμοποιηθεί σε διάφορες εφαρμογές όπως αναγνώριση εικόνας, αναγνώριση ομιλίας και επεξεργασία φυσικής γλώσσας. Αυτές οι δύο διαφοροποιήσεις είναι και το βασικό χαρακτηριστικό που ενδιαφέρει και την κοινότητα κυρίως για την δημιουργία μεθόδων. Στο πλαίσιο της εργασίας αυτής μας ενδιαφέρουν τα αδιαφανή μοντέλα και οι εξηγήσεις τους, όπου αυτά τα μοντέλα χρειάζονται post-hoc μεθόδους για να μπορέσουμε να δούμε τι ‘‘κρύβουν’’ και προς αυτά θα στηριχθούμε.

Η πιστότητα αναφέρεται στον βαθμό στον οποίο ο επεξηγητής μιμείται τον μηχανισμό λειτουργίας του μαύρου κουτιού που εξηγεί.

Όταν αναφερόμαστε σε παγκόσμιες μεθόδους εννοούμε ότι είναι μέθοδοι που μας δειχνουν τι γίνεται μέσα στο μοντέλο σαν οντότητα, δηλαδή μας βοηθούν να καταλάβουμε την θεμελιώδης δομή, τις υποθέσεις και της παραμέτρους που αυξάνουν την συνολική επιφάνεια των μηχανισμών λειτουργίας. Ενώ όταν μιλάμε για τις τοπικές μεθόδους εννοούμε ότι μας δίνει πληροφορίες σχετικά με κάθε πρόβλεψη, συσχετίζοντας την κάθε είσοδο του μοντέλου με την έξοδο που βγάζει προς τα έξω κάθε φορά.

Οι posthoc επεξηγήσεις είναι μια κατηγορία εξηγήσεων που προσπαθούν να προσεγγίσουν τον μηχανισμό του μαύρου κουτιού χωρίς να γίνει κάποια τροποποίηση στην αρχιτεκτονική ή στις παραμέτρους του μαύρου κουτιού. Ενώ οι antehoc επεξηγήσεις επιβάλλουν αλλαγές στο μαύρο κουτί, ώστε να αποκτήσουν την ικανότητα να μπορούν να εξηγηθούν από μόνες τους με τέτοιο τρόπο που να είναι ανάλογης ευκολίας όπως τα εγγενώς ερμηνεύσιμα μοντέλα.

Ώς αντιπαραστατικά αναφέρονται οι υποθετικές περιπτώσεις που κατευθύνουν την πρόβλεψη του μαύρου κουτιού προς την επιθυμητή κλάση ενδιαφέροντος. Επομένως οι αντιπαραστατικές εξηγήσεις (counterfactual explanations) αναφέρονται ώς οι οικογένειες των επεξηγηματικών μεθόδων που στοχεύουν στη δημιουργία υποθετικών αντιπαραστατικών που μεταβάλλουν την πρόβλεψη σε

μια επιθυμητή κλάση. Ενώ οι διαβουλευτικές εξηγήσεις στοχεύουν να εξάγουν χαρακτηριστικά εισόδου που βοηθούν να δικαιολογήσουν μια πρόβλεψη.

Οι οπτικές επεξηγήσεις προσπαθούν να εξάγουν τον τρόπο λειτουργίας ενός μαύρου κουτιού μέσω οπτικών ερεθισμάτων που να είναι σε τέτοια μορφή που μπορεί να κατανοήσει ο άνθρωπος. Ενώ οι κειμενικές εξηγήσεις αξιοποιούν την φυσική γλώσσα ώστε να περιγράψουν την λειτουργία του ταξινομητή.

Όταν αναφερόμαστε στην ερμηνευσιμότητα αναφερόμαστε στην ικανότητα ενός συστήματος να μπορεί να εξηγήσει ή να παρέχει ερμηνείες σε τέτοιους όρους ώστε να μπορεί να τους κατανοήσει ο άνθρωπος. Ενώ όταν αναφερόμαστε την επεξηγησιμότητα αναφερόμαστε στην έννοια της εξήγησης σαν διεπαφή ανάμεσα στον άνθρωπο και στο σύστημα λήψης αποφάσεων. Τέλος όταν αναφερόμαστε στην διαφάνεια εννοούμε να μπορεί να είναι από μόνο του κατανοητό, για παράδειγμα ένα δέντρο απόφασης θεωρείται από την κατασκευή του διαφανές.

2.4 Ταξινόμηση μεθόδων

Οι μέθοδοι της επεξηγήσιμης τεχνητης νοημοσύνης συχνά κατηγοριοποιούνται σε δύο μεγάλες οικογένειες, posthoc και antehoc, και η κατηγοριοποίηση αυτών των οικογενειών βασίζεται στο στάδιο στο οποίο οι εξηγήσεις ενσωματώνονται.

Οι posthoc τεχνικές προσπαθούν να αναπαράξουν εξηγήσεις χωρίς να γίνει κάποια τροποποίηση στην αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου, όμως μπορούν να προσπελάσουν τα ενδιάμεσα στρώματα του αν είναι απαραίτητο για αυτές. Εφόσον δεν γίνονται αλλαγές στην αρχιτεκτονική δεν υπάρχει κάποια ανάγκη για επανεκπαίδευση του μαύρου κουτιού, αυτο δίνει την δυνατότητα να δημιουργηθούν εξηγήσεις πάνω σε ήδη αναπτυγμένα μοντέλα. Ωστόσο η διασφάλιση της πιστότητας της παραγόμενης εξήγησης αποτελεί μια βασική πρόκληση κατα τη χρήση τέτοιων μεθόδων για την εξήγηση ενός συνελικτικού νευρωνικού δικτύου. Συγκεκριμένα η διασφάλιση της συνοχής μεταξύ της κατάταξης της εξήγησης των χαρακτηριστικών με βάση τη σημασία τους για την πρόβλεψη και της κατάταξης από το μαύρο κουτί που προσπαθεί να εξηγήσει, είναι μια μη τετριμμένη απαίτηση που πρέπει να πληροί μία τέτοια μέθοδος. Από την άλλη πλευρά, οι antehoc τεχνικές ενσωματώνουν την πτυχή της επεξήγησης και μεγιστοποιούν την ακρίβεια ταξινόμησης στο πλαίσιο της μάθησης. Το καταφέρνουν αυτό είτε επιρεάζοντας την αρχιτεκτονική του μαύρου κουτιού είτε προτείνουν νέες αρχιτεκτονικές όπου εντοπίζονται επεξηγήσιμα στοιχεία. Εφόσον η επεξηγησιμότητα είναι ενσωματωμένη στην διαδικασία της εκπαίδευσης, οι εξηγήσεις που δημιουργούνται πληρούν την πιστότητα προς το μοντέλο. Δηλαδή οι εξηγήσεις αποκαλύπτουν τον πραγματικό μηχανισμό που χρησιμοποιεί το συνελικτικό νευρωνικό δίκτυο ώστε να φτάσει σε μία πρόβλεψη. Με αυτήν την τεχνική όμως επανεκπαιδεύοντας ή τροποποιόντας την αρχιτεκτονική του μοντέλου μειώνεται η ακρίβεια του.

Οι μέθοδοι μπορούν να κατηγοριοποιηθούν βάση του πεδίου εφαρμογής (scope), είτε τοπικές (local) είτε παγκόσμιες (global), ανάλογα εάν οι επεξηγήσεις προσπαθούν να ερμηνεύσουν τους μηχανισμούς του μοντέλου συνολικά ή εάν ερμηνεύουν την συμπεριφορά του σε ένα συγκεκριμένο υποσύνολο περιπτώσεων. Οι παγκόσμιες μέθοδοι μπορούν να εξηγήσουν το πλήρες σύνολο κλάσεων και χρησιμοποιούνται ώστε να δημιουργήσουν ποιό εύκολα μοντέλα που θα μιμούνται το μαύρο κουτί. Τέτοια μοντέλα είναι χρήσιμα κυρίως σε εφαρμογές που θεωρούνται κρίσιμες και η επεξηγησιμότητα έχει πολύ σημαντικό ρόλο. Όμως η δημιουργία τέτοιον παγκόσμιων μεθόδων που αποτυπώνουν πιστά τα μη γραμμικά χαρακτηριστικά που έμαθε το συνελικτικό νευρωνικό δίκτυο είναι μεγάλη πρόκληση. Για την αντιμετώπιση της πρόκλησης αυτής, τοπικές επεξηγήσεις αξιοποιούν την τοπική γραμμικότητα για να εξηγήσουν το μοντέλο σε ένα συγκεκριμένο υπο σύνολο

περιπτώσεων. Μια προσέγγιση παγκόσμιας εξήγησης μπορεί να ληφθεί με τη συγκέντρωση τοπικών επεξηγήσεων στο σύνολο περιπτώσεων.

Ακόμα μπορούν να κατηγοριοποιηθούν σε model-specific και model-agnostic, ανάλογα τις υποθέσεις που έχουν επιλεχθεί για τον τύπο του μαύρου κουτιού. Οι model-specific μέθοδοι υποθέτουν ότι το μαύρο κουτί ακολουθεί συγκεκριμένη αρχιτεκτονική ώστε να καταφέρει να εξάγει τις εξηγήσεις για τον μηχανισμό του μοντέλου. Ενώ οι model-agnostic μέθοδοι εξάγουν τις εξηγήσεις τους αναλύοντας τις αλληλεπιδράσεις των δεδομένων εισόδου με τα δεδομένα εξόδου χωρίς να υποθέσουν τίποτα για την αρχιτεκτονική του μοντέλου.

Ένα ακόμη χαρακτηριστικό είναι ότι μπορούν να κατηγοριοποιηθούν βάση της κλάσης για την οποία ζητείται η εξήγηση. Συμβουλευτικές εξηγήσεις (Deliberative explanations) παρέχουν αποδείξεις για τις προβλέψεις του μαύρου κουτιού. Βοηθούν ώστε να αναγνωριστούν προκαταλήψεις στο μοντέλο και προσφέρουν πληροφορίες για την διαδικασία λήψης αποφάσεων. Ενώ οι αντιφατικές εξηγήσεις (counterfactual explanations) δίνουν την δυνατότητα επεξεργασίας ενός δοθέντος παραδείγματος προκειμένου να αλλαχθεί η προβλεπόμενη ετικέτα.

Φυσικά αυτο που είναι σημαντικό να αναφέρουμε αυτην την στιγμή είναι ότι οι μέθοδοι μπορούν να έχουν παραπάνω από ένα χαρακτηριστικά κατηγοριοποίησης που αναφέρθηκαν παραπάνω.

2.4.1 Posthoc μέθοδοι

Posthoc XAI μέθοδοι αναφέρονται σε τεχνικές και μεθόδους που χρησιμοποιούνται για να εξηγήσουν την συμπεριφορά των συστημάτων τεχνητής νοημοσύνης αφότου έχουν εκπαιδευτεί να κάνουν μια πρόβλεψη. Αυτές οι μέθοδοι δεν χρειάζεται απαραίτητα να τροποποιήσουν το σύστημα αλλά να αναλύσουν τις εξόδους του συστήματος για να παρέχουν εξηγήσεις για την διαδικασία λήψης αποφάσεων. Κύριο θετικό χαρακτηριστικό είναι ότι δεν απαιτεί αλλαγές στην αρχιτεκτονική του μοντέλου ή επανεκπαίδευση απλός χρησιμοποιούν το μοντέλο ως έχει ώστε να καταλάβουν πως δουλεύει. Οι posthoc μέθοδοι μπορούν να κατηγοριοποιηθούν στις εξής περετέρο υποκατηγορίες: Saliency Map, Model-agnostic, Counterfactual.

2.4.1.1 Χάρτες ενεργοποίησης τάξης (Class activation maps)

Η πιό κοινή τεχνική για να εξηγήσουμε τα συνελικτικά νευρωνικά δίκτυα είναι να αναγνωρίσουμε τις περιοχές εκείνες που συνεισφέρουν περισσότερο στην πρόβλεψη. Οι περιοχές αυτές συνήθως τις εμφανίζουμε χρησιμοποιώντας τα saliency maps, όπου η εικόνα θα χρωματιστεί βάση της σημαντικότητας των περιοχών.

Οι χάρτες ενεργοποίησης τάξης (class activation maps) υποθέτουν ότι η περιοχή που είναι πιο σχετική για την πρόβλεψη μπορεί να αποκτηθεί από έναν σταθμισμένο συνδυασμό των χαρτών ενεργοποίησης απο τα φίλτρα του συνελικτικού επιπέδου. Οι περισσότερες μέθοδοι εξάγουν τους χάρτες ενεργοποίησης απο το τελευταίο συνελικτικό επίπεδο που είναι πιο κοντά στην έξοδο, διότι έχει παρατηρηθεί πως τα τελευταία επίπεδα κωδικοποιούν πολύπλοκα μέρη. Οι αλγόριθμοι επεξήγησης εκτιμούν τις περιοχές τις οποίες το νευρωνικό δίκτυο ενδιαφέρεται περισσότερο μέσω ενός χάρτη προεξοχής (saliency map) το οποίο μπορεί να εκφραστεί ως ένας σταθμισμένος συνδυασμός των χαρτών ενεργοποίησης από καθένα από τα φίλτρα. Ο χάρτης προεξοχής χαμηλών διαστάσεων που λαμβάνεται μέσω του σταθμισμένου συνδυασμού των χαρτών ενεργοποίησης από τα μεμονωμένα φίλτρα, αναδεικνύεται στη συνέχεια σε πλήρες μέγεθος εικόνας για να δημιουργηθεί μια εξήγηση που δείχνει την περιοχή της εικόνας στην οποία εστιάζει το νευρωνικό δίκτυο για να καταλήξει στην πρόβλεψη. Διάφοροι μηχανισμοί έχουν προταθεί για την εκτίμηση των βαρών που συνδυάζουν τους χάρτες ενεργοποίησης από τα φίλτρα. Αυτές οι προσεγγίσεις μπορούν να διακλαδιστούν με βάση των κλίσεων.

Οι διαβαθμίσεις (gradients) καταγράφουν την κατεύθυνση κατά την οποία αυξάνεται η τιμή μιας συνάρτησης. Έτσι οι διαβαθμίσεις που διαδίδονται πίσω στα συνελικτικά στρώματα από το επίπεδο εξόδου φέρουν ένα σήμα που υποδεικνύει τα χαρακτηριστικά των οποίων η παρουσία οδηγεί το μοντέλο προς την επιθυμητή πρόβλεψη. Αυτό το σήμα αξιοποιείται για την εκτίμηση των βαρών και τον συνδυασμό των χαρτών ενεργοποίησης χρησιμοποιώντας προσεγγίσεις προεξοχής που βασίζονται σε κλίση. Το Grad-CAM (Gradient-weighted Class Activation Mapping) δημιουργεί έναν χάρτη που επισημαίνει τις περιοχές της εικόνας εισόδου που ήταν πιο σχετικές με την πρόβλεψη του νευρωνικού δικτύου. Λειτουργεί με τον υπολογισμό των διαβαθμίσεων της πρόβλεψης εξόδου σε σχέση με τις ενεργοποιήσεις του τελικού συνελικτικού στρώματος. Οι χάρτες ενεργοποίησης συνδυάζονται με βάση τα βάρη που λαμβάνονται με τον μέσο όρο των κλίσεων σε σχέση με το αντίστοιχο φίλτρο σε όλες τις χωρικές θέσεις. Δεν απαιτούνται πρόσθετες τροποποιήσεις στην αρχιτεκτονική των νευρωνικών δικτύων για τη δημιουργία επεξηγήσεων και επομένως μπορούν να χρησιμοποιηθούν για να εξηγήσουν οποιοδήποτε συνελικτικό νευρωνικό δίκτυο. Οι ενσωματωμένες διαβαθμίσεις (Integrated gradients) θεωρούν μια είσοδο αναφοράς και διασχίζουν τον χώρο του στιγμιότυπου κατά μήκος της διαδρομής από μια είσοδο αναφοράς για να φτάσουν στο δεδομένο στιγμιότυπο. Οι αποδόσεις σε σχέση με τα ενδιάμεσα στιγμιότυπα κατά μήκος της διαδρομής ενσωματώνονται για να ληφθεί ένας ισχυρός χάρτης προεξοχής που απεικονίζει τα εμφανή pixel στη δεδομένη εμφάνιση. Η οπίσθια διάδοση διέγερσης (backpropagation) χρησιμοποιεί μια πιθανολογική στρατηγική κερδοφορίας όπου η απόδοση που διαδίδεται σε έναν νευρώνα καθορίζεται πιθανολογικά. Η καθοδηγούμενη οπίσθια διάδοση (Guided backpropagation) προτείνει τη διάδοση της απόδοσης μόνο σε εκείνους τους νευρώνες που ήταν ενεργοί κατά τη διάρκεια του μπροστινού περάσματος, δημιουργώντας έτσι λεπτότερους χάρτες εξέχουσας θέσης σε επίπεδο pixel σε σύγκριση με την οπίσθια διάδοση (backpropagation) που εξάπλωνε τις διαβαθμίσεις ως απόδοση ανεξάρτητα από τη συμβολή του νευρώνα μέχρι να φτάσει στο στρώμα εξόδου.

Layerwise Relevance Propagation διαδίδει την έξοδο νευρωνικού δικτύου πίσω μέσω των διαφορετικών επιπέδων για να εκχωρήσει βαθμολογίες συνάφειας σε αυτά τα χαρακτηριστικά εισόδου. Το μπροστινό πέρασμα διαδίδει την ενεργοποίηση από το επίπεδο εισόδου και φτάνει στο επίπεδο εξόδου. Η διάδοση της συνάφειας ξεκινά προς την αντίθετη κατεύθυνση από το επίπεδο εξόδου και σταδιακά το σήμα συνάφειας φτάνει στα μεμονωμένα εικονοστοιχεία εισόδου. Το σήμα συνάφειας από έναν νευρώνα κατανέμεται σε όλους τους νευρώνες που έχουν συνεισφέρει σε αυτό κατά τη διάρκεια της διέλευσης προς τα εμπρός ανάλογα με τη συμβολή τους. Το Deep-LIFT είναι μια τροποποιημένη μορφή διάδοσης συνάφειας όπου οι διαφορές μεταξύ ενεργοποιήσεων σε σχέση με μια είσοδο αναφοράς διαδίδονται για να ληφθεί η συνάφεια των διαφορετικών χαρακτηριστικών εισόδου.

Διάφορες ποσοτικές μετρήσεις έχουν προταθεί για την αξιολόγηση της πιστότητας των δημιουργούμενων εξηγήσεων. Οι προτεινόμενες μετρήσεις βασίζονται στην απαίτηση ότι η αφαίρεση μιας σημαντικής περιοχής πρέπει να μειώνει την εμπιστοσύνη πρόβλεψης του μοντέλου ενώ η παρουσία της πρέπει να ενισχύει την εμπιστοσύνη

2.4.1.2 Model-agnostic εξηγήσεις

Οι μέθοδοι model-agnostic αναφέρονται στις οικογένειες των μεθόδων, οι οποίες εξηγούν τη λειτουργία ενός μοντέλου μαύρου κουτιού παρατηρώντας τις αλληλεπιδράσεις εισόδου-εξόδου. Μπορούν να εφαρμοστούν σε οποιοδήποτε μοντέλο μηχανικής μάθησης, ανεξάρτητα από τον τύπο ή την αρχιτεκτονική του, και μπορούν να λειτουργήσουν για να εξηγήσουν δεδομένα οποιασδήποτε μορφής, όπως κείμενο και εικόνες. Το πεδίο εφαρμογής αυτών των επεξηγήσεων των μεθόδων μπορεί να είναι τοπικό σε ένα δεδομένο ή μπορεί να εξηγήσει συνολικά τη συνολική λειτουργία του μαύρου κουτιού. Αυτές οι μέθοδοι στοχεύουν στην κατασκευή ενός εγγενώς ερμηνεύσιμου ψευδο ταξινομητή

που προσεγγίζει τον μηχανισμό λειτουργίας του ταξινομητή μαύρου κουτιού που θα εξηγηθεί είτε τοπικά γύρω από μια μικρή γειτονιά ενός στιγμιότυπου για το οποίο αναζητείται η εξήγηση, είτε συνολικά που εκτείνεται σε ολόκληρο τον χώρο παρουσίας του ταξινομητή.

Το Local Interpretable Model-agnostic Explanations (LIME) δημιουργεί ένα απλούστερο, πιο ερμηνεύσιμο μοντέλο, για παράδειγμα, έναν γραμμικό παλινδρομητή ή ένα δέντρο αποφάσεων του οποίου η πολυπλοκότητα είναι βελτιστοποιημένη έτσι ώστε ο προσδιορισμένος ψευδο ταξινομητής να μιμείται τη συμπεριφορά του αρχικού μοντέλου στην τοπική γειτνίαση της εισόδου γύρω από το στιγμιότυπο που πρέπει να εξηγηθεί. Αυτό το απλούστερο μοντέλο μπορεί στη συνέχεια να χρησιμοποιηθεί για την παροχή τοπικών εξηγήσεων για μεμονωμένες προβλέψεις. Μπορεί να παρατηρηθεί ότι μπορούν να δημιουργηθούν διαφορετικές εξηγήσεις για την ίδια περίπτωση, ανάλογα με τους γείτονες του δείγματος βάσει των οποίων εκτιμάται η τοπική γειτονιά. Οι Anchors δημιουργούν εξηγήσεις για μεμονωμένες προβλέψεις χρησιμοποιώντας κανόνες if–then κατασκευασμένους με τρόπο από κάτω προς τα πάνω, έτσι ώστε ο κανόνας να καλύπτει επακριβώς τους τοπικούς γείτονες του στιγμιότυπου που πρόκειται να εξηγηθεί. Το MAIRE επεκτείνει τα Anchors για να χειρίζεται χαρακτηριστικά συνεχούς αξίας μαθαίνοντας να κατασκευάζει ένα βέλτιστο ορθότοπο αυτόματα, σε αντίθεση με την προηγούμενη προσέγγιση που απαιτεί το εύρος τιμών για την κατασκευή του ορθότοπου. Οι τοπικές μέθοδοι επεξήγησης στοχεύουν στην εξαγωγή επεξηγήσεων που είναι πιστές σε μια τοπική γειτονιά μέσω ειδικών μετρήσεων, όπως η κάλυψη που εκτιμά το κλάσμα των περιπτώσεων που βρίσκονται κοντά στον επεξηγητή και η ακρίβεια που υποδηλώνει το κλάσμα των καλυπτόμενων περιπτώσεων των οποίων η πρόβλεψη από τον επεξηγητή ταιριάζει με την πρόβλεψη του μαύρου κουτιού. Η κατασκευή ενός επεξηγητή MAIRE μεγιστοποιεί την κάλυψη, διασφαλίζοντας πιστότητα στο υποκείμενο μαύρο κουτί ικανοποιώντας ένα επίπεδο ακρίβειας που έχει ορίσει ο χρήστης. Αν και αυτές οι μέθοδοι προσφέρουν τοπικές εξηγήσεις, μια συνολική κατανόηση του μοντέλου μπορεί να επιτευχθεί μόνο με τη συγκέντρωση των τοπικών εξηγήσεων σε ένα σύνολο περιπτώσεων.

Έχουν επίσης γίνει προσπάθειες να δημιουργηθεί μια επεξήγηση που να προσεγγίζει την παγκόσμια συμπεριφορά του μοντέλου στο σύνολό του. Το SHAP χρησιμοποιεί τις αρχές από τη θεωρία παιγνίων (Shapley values) για να εκχωρήσει μια βαθμολογία σημασίας σε κάθε χαρακτηριστικό εισόδου, υποδεικνύοντας πόσο συνεισφέρει κάθε χαρακτηριστικό στην έξοδο του συστήματος. Αυτές οι βαθμολογίες σπουδαιότητας μπορούν να χρησιμοποιηθούν για τον προσδιορισμό των πιο σχετικών χαρακτηριστικών και την κατανόηση της επιρροής τους στις αποφάσεις του συστήματος. Ο υπολογισμός των τιμών Shapley απαιτεί την εξέταση όλων των πιθανών υποσυνόλων του χώρου χαρακτηριστικών και την αξιολόγηση της επίδρασης διαταραχής κάθε υποσυνόλου στην έξοδο. Αυτό είναι υπολογιστικά εξαντλητικό λόγω της εκθετικής χρονικής πολυπλοκότητας. Πολλές προσεγγίσεις έχουν προταθεί με βάση τις τιμές Shapley που προσεγγίζονται λαμβάνοντας υπόψη μόνο τη διαταραχή ενός χαρακτηριστικού τη φορά. Η σημασία του χαρακτηριστικού μετάθεσης υπολογίζει τη σημασία κάθε χαρακτηριστικού εισόδου μεταθέτοντας τυχαία τις τιμές του και μετρώντας τη μείωση της απόδοσης του μοντέλου. Τα διαγράμματα μερικής εξάρτησης οπτικοποιούν τη σχέση μεταξύ ενός χαρακτηριστικού εισόδου και της πρόβλεψης του μοντέλου διατηρώντας όλα τα άλλα χαρακτηριστικά σταθερά.

Μια άλλη σημαντική παρατήρηση είναι ότι οι μέθοδοι αγνωστικοποίησης μοντέλων αναπτύσσονται για να δημιουργήσουν εξηγήσεις για οποιοδήποτε μοντέλο μαύρου κουτιού, και ως εκ τούτου δεν γίνεται καμία υπόθεση σχετικά με την αρχιτεκτονική του. Η εξήγηση δίνεται ως προς τα χαρακτηριστικά εισόδου που είναι σημαντικά για την πρόβλεψη. Στις εικόνες, τα pixel αποτελούν τα χαρακτηριστικά εισόδου. Καθώς οι επεξηγήσεις σε επίπεδο εικονοστοιχείων δεν είναι εύκολα ερμηνεύσιμες για τους ανθρώπους, μια λύση προτείνεται να χρησιμοποιηθεί μια συλλογή από χωρικά πιο κοντινά εικονοστοιχεία που ονομάζονται superpixel. Αυτά τα superpixel χρησιμεύουν ως σύνθετα

χαρακτηριστικά εισόδου για τις μεθόδους αγνωστικοποίησης μοντέλων για τη δημιουργία επεξηγήσεων.

2.4.1.3 Αντιπαραστατικές εξηγήσεις (Counterfactual explanations)

Οι αντιπαραστατικές εξηγήσεις περιλαμβάνουν τη δημιουργία εναλλακτικών σεναρίων για να εξηγηθεί η συμπεριφορά ενός συστήματος. Αυτές οι αντιπαραστατικές εξηγήσεις μπορούν να βοηθήσουν τους χρήστες να κατανοήσουν τη διαδικασία λήψης αποφάσεων και να εντοπίσουν πιθανές προκαταλήψεις ή σφάλματα στο σύστημα. Διαφέρουν από τις διαβουλευτικές εξηγήσεις που στοχεύουν να δικαιολογήσουν γιατί έγινε μια συγκεκριμένη πρόβλεψη. Οι επεξηγήσεις αντιπαραστατικών προχωρούν ένα βήμα παραπέρα για να αναλύσουν τις αλλαγές στην είσοδο για να λάβουν μια άλλη επιθυμητή πρόβλεψη. Αυτός ο τρόπος εξήγησης μπορεί να εφαρμοστεί για την ανάλυση ενός ταξινομητή που λειτουργεί με οποιαδήποτε μορφή δεδομένων, είτε είναι πίνακας, κείμενο ή εικόνα. Οι μέθοδοι προσπαθούν να πραγματοποιήσουν ελάχιστες τροποποιήσεις στο δεδομένο ερώτημα, έτσι ώστε η πρόβλεψη να κατευθύνεται προς μια εναλλακτική επιθυμητή κλάση. Αυτό μπορεί να θεωρηθεί ως διαταραχές που σκοπεύουν να ανατρέψουν την πρόβλεψη. Στην περίπτωση των δεδομένων πινάκων, όπου η αποτελεσματικότητα των αντιπαραστατικών προσεγγίσεων έχει ως επί το πλείστον αποδειχθεί, οι διαταραχές είναι διαχειρίσιμες καθώς είναι γνωστό το εύρος τιμών που μπορούν να λάβουν τα χαρακτηριστικά του πίνακα και το παράδειγμα μπορεί να διαταραχθεί για να δημιουργήσει ένα άλλο ρεαλιστικό παράδειγμα που βρίσκεται εντός της πολλαπλότητας στην οποία εκπαιδεύτηκε ο ταξινομητής. Ο προσδιορισμός αυτής της ρεαλιστικής πολλαπλότητας δεν είναι τετριμμένος στην περίπτωση εικόνων των οποίων τα συστατικά, γνωστά και ως τα pixel, μπορούν θεωρητικά να λάβουν οποιαδήποτε πραγματική τιμή. Ο στόχος της εξήγησης με χρήση μιας διαταραγμένης παρουσίας είναι κοινός στην αντίθετη μάθηση (adversarial learning), εκτός από το ότι δεν έχει μια κατηγορία στόχο προς την οποία πρέπει να κατευθυνθεί η πρόβλεψη. Ο στόχος στη δημιουργία ενός αντιπάλου παραδείγματος είναι ότι η πρόβλεψη για το παραγόμενο στιγμιότυπο δεν πρέπει να είναι ίδια με αυτή του μη διαταραγμένου στιγμιότυπου. Πρέπει να τηρείται προσοχή καθώς μια τυχαία διαταραχή μπορεί να δημιουργήσει ένα αντίθετο παράδειγμα, το οποίο μπορεί να ανατρέψει μια πρόβλεψη προς τον στόχο ενδιαφέροντος, αλλά μπορεί να μην είναι ιδανικός υποψήφιος για εξαγωγή αντιπαραστατικών εξηγήσεων, καθώς το παράδειγμα μπορεί να είναι ακραίο σε σχέση με τη ρεαλιστική εκπαίδευση πολλαπλότητας εικόνων, αμφισβητώντας έτσι την πιστότητα της παραγόμενης αντιπαραστατικής εξήγησης στο υποκείμενο μοντέλο και δεδομένα. Για να παρακάμψουν αυτήν την πρόκληση, οι υπάρχουσες προσεγγίσεις είτε διατηρούν μια τράπεζα εικόνων από την οποία επιλέγεται η πλησιέστερη αντίθετη εικόνα είτε χρησιμοποιείται ένα παραγωγικό μοντέλο για τη δειγματοληψία των αντιπραγματικών γειτόνων του στιγμιότυπου ερωτήματος από τη διανομή στην οποία εκπαιδεύεται το συνελικτικού νευρωνικού δικτύου.

Υπήρξαν επίσης ορισμένες προσεγγίσεις αντιπαραστατικής εξήγησης που επιτρέπουν την αναζήτηση επεξήγησης σε σχέση με μια άλλη κατηγορία ενδιαφέροντος, η οποία μπορεί να αξιοποιηθεί για να δημιουργήσει μια αντίθετη εξήγηση για την εναλλακτική κατηγορία ενδιαφέροντος στόχου. Ωστόσο, αυτές οι προσεγγίσεις δεν δημιουργούν επεξηγήσεις που ποικίλλουν σημαντικά σε σχέση με την εναλλακτική κλάση ερωτήματος.

Η αρχική προσέγγιση για τη δημιουργία αντιπραγματικών επεξηγήσεων μέσω ρεαλιστικών περιπτώσεων είναι η διατήρηση μιας τράπεζας εικόνων από την οποία επιλέγεται η πλησιέστερη αντίθετη περίπτωση σε μια δεδομένη περίπτωση δοκιμής. Διάφορες προσεγγίσεις έχουν εξετάσει διαφορετικούς τρόπους εκτίμησης του πλησιέστερου στιγμιότυπου. Οι Wang & Vasconcelos [48] δημιουργούν σκόπιμες επεξηγήσεις για τη δεδομένη περίπτωση δοκιμής και όλες τις παρουσίες στην τράπεζα αντιπραγματικών εικόνων και επιλέγουν την παρουσία που περιέχει χαρακτηριστικά που υποστηρίζουν την αντίθετη κατηγορία και καμία πληροφορία της προβλεπόμενης κλάσης ως την

πλησιέστερη αντίθετη περίπτωση. Οι Goyal et al. [48] προσομοιώνει τους χάρτες χαρακτηριστικών μετάθεσης για να αποκτήσουν χαρακτηριστικά πιο κοντά σε εκείνα των αντίθετων περιπτώσεων που κατευθύνουν την πρόβλεψη προς την επιθυμητή κλάση. Ένας κύριος περιορισμός αυτών των προσεγγίσεων είναι η ανάγκη να περάσουμε από την τράπεζα εικόνων για κάθε δοκιμαστική περίπτωση που πρέπει να εξηγηθεί. Επιπλέον, η τράπεζα εικόνων πρέπει να δημιουργηθεί δειγματοληπτικά από την ίδια διανομή με τα δεδομένα στα οποία έχει εκπαιδευτεί το συνελικτικό νευρονικό δίκτυο.

Για να διατηρηθεί η διανομή, ένα εναλλακτικό σύνολο προσεγγίσεων χρησιμοποίησε παραλλαγές των Generative Adversarial Networks (GAN) για να μάθει την υποκείμενη διανομή. Ωστόσο, πρέπει να σημειωθεί ότι τα παραγωγικά μοντέλα που χρησιμοποιούνται για την εκμάθηση της υποκείμενης κατανομής είναι, και πάλι, μαύρα κουτιά των οποίων η λειτουργία είναι άγνωστη. Αυτό περιπλέκει το υπό εξέταση πρόβλημα καθώς οι τεχνικές για την ερμηνεία του GAN πρέπει να χρησιμοποιηθούν πέρα από τις υπάρχουσες αντιπαραστατικές επεξηγήσεις.

2.4.2 Antehoc μέθοδοι

Η επεξήγηση Antehoc, ή η επεξήγηση βάσει σχεδίου, όπως αποκαλείται ευρέως, αναφέρεται στην πρακτική της κατασκευής συστημάτων τεχνητής νοημοσύνης με γνώμονα την επεξήγηση και την ερμηνευσιμότητα από την αρχή και όχι ως εκ των υστέρων. Με την ενσωμάτωση της επεξήγησης στη διαδικασία σχεδιασμού, αυτές οι μέθοδοι στοχεύουν στη δημιουργία συστημάτων τεχνητής νοημοσύνης που είναι εγγενώς διαφανή, ερμηνεύσιμα και αξιόπιστα. Παρά τα πλεονεκτήματα όπως η εγγενής ερμηνευτικότητα και η αξιοπιστία που μπορούν να προσφέρουν οι προηγούμενες επεξηγήσεις, ο σχεδιασμός τέτοιων μοντέλων μπορεί να είναι δύσκολος και μπορεί να απαιτεί γνώση και εξειδίκευση σε συγκεκριμένο τομέα. Επιπλέον, ορισμένες μέθοδοι ερμηνευσιμότητας μπορεί να βαρύνουν την απόδοση του μοντέλου, περιορίζοντας τη χρησιμότητά τους σε ορισμένες εφαρμογές. Για να ενσωματωθεί η δυνατότητα επεξήγησης, η αρχιτεκτονική των υπαρχουσών αρχιτεκτονικών CNN πρέπει να τροποποιηθεί ή να επινοηθούν νέα στοιχεία που μπορούν να ερμηνευθούν από το σχεδιασμό. Η εξήγηση μπορεί να είναι η επισήμανση οπτικών τεχνουργημάτων που οδηγούν στην πρόβλεψη ή η παροχή περιγραφών κειμένου που δικαιολογούν τις προβλέψεις.

2.4.2.1 Οπτικοποιημένες εξηγήσεις (Visual explanations)

Παρόμοια με τον τρόπο με τον οποίο τα συνελικτικά νευρωνικά δίκτυα έμαθαν να εξάγουν χαρακτηριστικά αυτόματα από τα δεδομένα, η κοινότητα πρότεινε να επιβάλει στα συνελικτικά νευρωνικά δίκτυα να μαθαίνουν ερμηνεύσιμες έννοιες αυτόματα από τα δεδομένα και να τις χρησιμοποιούν για να προβλέψουν την κατηγορία αντικειμένων. Οι διακριτικές ερμηνεύσιμες έννοιες μαθαίνονται αυτόματα από τα δεδομένα και η ανίχνευση αυτών των εννοιών σε δοκιμαστικές περιπτώσεις καθοδηγεί την πρόβλεψη χρησιμοποιώντας έναν εγγενώς ερμηνεύσιμο παράγοντα πρόβλεψης, όπως ένας γραμμικός παλινδρομητής ή ένα δέντρο αποφάσεων, επιτρέποντας την αποκάλυψη του πλήρους συλλογισμού του τροποποιημένου συνελικτικού νευρωνικού δικτύου. Σε τέτοια μοντέλα, η ικανότητα παροχής εξηγήσεων ενσωματώνεται στη φάση της εκπαίδευσης από το σχεδιασμό.

Οι πρώτες οπτικές επεξηγηματικές προσεγγίσεις χρησιμοποιούσαν την προσοχή (attention), η οποία είναι μια επιλεκτική διατήρηση των χαρακτηριστικών για την ταξινόμηση του στιγμιοτύπου δοκιμής. Η προσοχή μπορεί να είναι σκληρή ή μαλακή με την έννοια ότι η επιλογή των περιοχών από τα χαρακτηριστικά μπορεί να είναι ντετερμινιστική ή πιθανολογική. Οι περιοχές που συμμετείχαν θα παραδοθούν ως εξήγηση. Ωστόσο, έχουν υπάρξει παρατηρήσεις ότι ένας οπτικοποιημένος χάρτης προσοχής δεν χρειάζεται να είναι μια ιδανική εξήγηση.

Καθώς η ικανότητα εξήγησης έχει ενσωματωθεί κατά τη φάση της εκπαίδευσης και το CNN καθοδηγείται να χρησιμοποιήσει αυτά τα εξηγήσιμα στοιχεία για να κάνει προβλέψεις, η πιστότητα αυτών των εξηγήσεων είναι εγγυημένη. Με άλλα λόγια, οποιαδήποτε πληροφορία αποκαλύπτει η εξήγηση είναι πραγματικά αυτή που χρησιμοποιεί το μοντέλο για να καταλήξει στην πρόβλεψη. Ωστόσο, πρέπει να επανεκπαιδευτεί από το μηδέν για να ενσωματώσει μια τέτοια επεξήγηση σε ένα συνελικτικό νευρωνικό δίκτυο. Αυτή η προοπτική μπορεί να αξιοποιηθεί όταν το μοντέλο δεν έχει ακόμη αναπτυχθεί, και είναι επιθυμητό να αναπτυχθεί ένα μοντέλο που μπορεί να εξηγηθεί αλλά δεν μπορεί να χρησιμοποιηθεί για ένα ήδη ανεπτυγμένο μοντέλο.

2.4.2.2 Εξηγήσεις φυσικής γλώσσας (Natural language explanations)

Οι προσεγγίσεις επεξήγησης φυσικής γλώσσας στοχεύουν στη δημιουργία περιγραφών κειμένου που παρέχουν πληροφορίες για το πώς ένας ταξινομητής εικόνων κάνει τις προβλέψεις του. Η βασική ιδέα πίσω από αυτήν την προσέγγιση είναι να αξιοποιήσει τις τεράστιες ποσότητες γλωσσικής γνώσης που έχει συσσωρευτεί κατά τη διάρκεια αιώνων χρήσης της γλώσσας και να την ενσωματώσει στο μοντέλο. Αυτό μπορεί να βοηθήσει το μοντέλο να δημιουργήσει πιο συνεκτικές και φυσικές εξηγήσεις που μπορούν να ερμηνεύσουν οι άνθρωποι.

Αυτή η προσέγγιση προϋποθέτει τη διαθεσιμότητα περιγραφής φυσικής γλώσσας για τις υπό εξέταση τάξεις και για μεμονωμένες περιπτώσεις από τις οποίες μπορεί να προσδιοριστεί η αντιστοίχιση μεταξύ οπτικών πτυχών και φράσεων φυσικής γλώσσας. Ένα εκπαιδευμένο γλωσσικό μοντέλο ενσωματώνεται για να λειτουργεί ως επεξηγητής στον αγωγό ταξινόμησης για την κατασκευή ενός συνελικτικού νευρωνικού δικτύου που μπορεί να δικαιολογήσει τη λειτουργία του μέσω φράσεων φυσικής γλώσσας. Τα οπτικά χαρακτηριστικά που εξάγονται από τον εξαγωγέα χαρακτηριστικών του συνελικτικού νευρωνικού δικτύου τροφοδοτούνται στο μοντέλο γλώσσας, το οποίο είναι εκπαιδευμένο να δημιουργεί λεζάντες που περιγράφουν το περιεχόμενο της εικόνας. Στη συνέχεια, μια μονάδα κριτικής αξιολογεί την ορθότητα της λεζάντας που δημιουργείται στο περιεχόμενο της εικόνας. Για την εκπαίδευση της ενότητας κριτικής, τα ζεύγη βασικής αλήθειας (εικόνα, λεζάντα) τυχαιοποιούνται και το μοντέλο εκπαιδεύεται ώστε να παρέχει χαμηλή βαθμολογία για μια τυχαία περίπτωση όπου η εικόνα και η λεζάντα δεν συμφωνούν και μια υψηλή βαθμολογία σε αληθινές περιπτώσεις όπου η εικόνα και οι λεζάντες συμφωνούν. Τα οπτικά χαρακτηριστικά και οι δημιουργημένοι υπότιτλοι από τη δοκιμαστική εικόνα τροφοδοτούνται στη μονάδα κριτικής, η οποία εξάγει μία βαθμολογία που υποδηλώνει την ποιότητα της λεζάντας που δημιουργείται. Για να αποφευχθούν πολλαπλά περάσματα ανάμεσα στο CNN και της γεννήτριας υποτίτλων με βάση την ανατροφοδότηση από τη μονάδα κριτικής, λαμβάνονται υπόψη οι κορυφαίοι υπότιτλοι από τη γεννήτρια υποτίτλων και η λεζάντα με την κορυφαία κατάταξη από τον κριτικό μεταβιβάζεται στην τοπική μονάδα για εντοπισμό της αντίστοιχης περιοχής εικόνας που συμβάλλει στη δημιουργία της λεζάντας.

Η προσέγγιση χρησιμοποιείται ως επί το πλείστον για να δικαιολογήσει τις προβλέψεις που γίνονται σε σχετικές εργασίες όρασης υπολογιστή, ειδικές εργασίες γλώσσας όρασης, όπως λεζάντες εικόνων, οπτική απάντηση σε ερωτήσεις.

Ωστόσο, ο σχεδιασμός αποτελεσματικών προσεγγίσεων επεξήγησης φυσικής γλώσσας μπορεί να είναι δύσκολος και μπορεί να απαιτεί γνώση και εξειδίκευση σε συγκεκριμένο τομέα. Επιπλέον, η ποιότητα και η αποτελεσματικότητα των δημιουργούμενων επεξηγήσεων μπορεί να ποικίλλει ανάλογα με την πολυπλοκότητα και την ακρίβεια του υποκείμενου ταξινομητή εικόνας και την ποιότητα των διαθέσιμων γλωσσικών σχολιασμών. Μια άλλη βασική πρόκληση που πρέπει να αντιμετωπιστεί κατά την ενσωμάτωση επεξηγήσεων φυσικής γλώσσας είναι ότι το γλωσσικό μοντέλο που διευκολύνει την αιτιολόγηση της πρόβλεψης είναι ένα άλλο μαύρο κουτί του οποίου ο μηχανισμός λειτουργίας πρέπει να αποκαλυφθεί.

3 Σχεδιασμός και υλοποίηση συστήματος

Σε αυτό το κεφάλαιο θα αναφερθούν τα εργαλεία τα οποία χρησιμοποιήθηκαν για το πειραματικό μέρος, καθώς και θα γίνει μια συζήτηση για τις τεχνικές με τις οποίες αποφασίστηκε να γίνει το πείραμα, θα αναλυθεί το σύνολο δεδομένων με το οποίο έγινε το πείραμα και την προεπεξεργασία που χρειάστηκε να υποστεί το σύνολο δεδομένων.

Για το πειραματικό κομμάτι έγινε χρήση του εργαλείου Pytorch, ένα εργαλείο, ανοιχτού κώδικα, το οποίο βοηθάει στην εκπαίδευση και στην δοκιμή ενός νευρωνικού δικτύου. Το αρνητικό το οποίο μπορεί να παρατηρήσει κατευθείαν κάποιος είναι ότι δεν διαθέτει προς χρήση δική του σουίτα με μεθόδους για τον σχεδιασμό των απαρέτητων plots που θα χρειαστεί κάποιος ερευνητής / μηχανικός. Κάποιος που θέλει να πειραματιστεί μαζί του θα χρειαστεί να βρει μια βιβλιοθήκη που είναι κατάλληλη για τις δικές του απαιτήσεις. Στα πλαίσια της διπλωματικής εργασίας τα plots που χρειάστηκε να δημιουργήσουμε, τα δημιουργήσαμε είτε μέσω έτοιμου εργαλείου που δίνει κάθε μέθοδος το οποίο αναγνωρίζει την δομή της επεξήγησεις και τα δείχνει κατάλληλα, είτε με την βοήθεια της βιβλιοθήκης matplotlib, μια βιβλιοθήκη που είναι αρκετά εύκολη και χρήσιμη όταν κάποιος θέλει να δημιουργήσει εύκολα και γρήγορα μερικά plots δίνοντας του αρκετές δυνατότητες. Όσων αφορά την επεξεργασία των δεδομένων δεν υπάρχουν και πολλά πράγματα που μπορεί να κάνει το εργαλείο Pytorch παρά μόνο μερικές τροποποιήσεις στα δεδομένων κατα την διάρκεια ανάκτησης τους. Τροποποιήσεις που αρκετά συχνά είναι χρήσιμες αλλά για όποια μετέπητη επεξεργασία των δεδομένων θα πρέπει ο ερευνητής να στηριχθεί σε εργαλεία όπως το Numpy.

Οι τεχνικές που χρησιμοποιήθηκαν για την εξαγωγή των εξηγήσεων του συνελικτικού νευρωνικού δικτύου είναι η τεχνική LIME και η τεχνική Deep SHAP. Δύο τεχνικές που έχουν συζητηθεί σε μεγάλο βαθμό από την κοινότητα και φαίνεται να έχουν χρησιμοποιηθεί και σε διάφορες εφαρμογές για να εξηγήσουν τα αποτελέσματα διάφορων μοντέλων. Η συζήτηση και το ενδιαφέρον γύρω από αυτές τις τεχνικές έπαιξαν σημαντικό ρόλο στην επιλογή τους, καθώς και η προσβασιμότητα τους απο τους διάφορους παρόχους βιβλιοθηκών. Αναφορικά με την προσβασιμότητα τους είναι χρήσιμο να αναφερθεί ότι μπορούν να ανακτηθούν οι τεχνικές μέσω του διαχειριστή πακέτων pip της Python αλλά κάποιος που χρησιμοποιεί το περιβάλλον ανάπτυξης λογισμικού Anaconda μπορεί να τα βρεί διαθέσιμα με τις κατάλληλες εντολές conda, μια σημαντική λεπτομέρεια καθώς είναι ένα προϊόν που χρησιμοποιείται αρκετά στην ανάπτυξη των data science εφαρμογών.

3.1 Λίγα λόγια για το Pytorch

Όπως αναφερθήκαμε και παραπάνω το εργαλείο Pytorch είναι ανοιχτού κώδικα και βοηθάει στην εκπαίδευση και δοκιμή ενός νευρωνικού δικτύου. Το νευρωνικό δίκτυο μπορεί είτε να δημιουργηθεί από τον χρήστη / ερευνητή για να καλύψει τις ανάγκες του, είτε μπορεί να ανακτηθεί κάποιο προεκπεδευμένο δίκτυο πάνω σε δεδομένα που τον ενδιαφέρουν. Το να βρεί κάποιος ένα προεκπεδευμένο δίκτυο είναι λίγο δύσκολο καθώς θα πρέπει να είναι ένα πολύ γνωστό σύνολο δεδομένων, το οποίο όμως θεωρείται και χρήσιμο από την κοινότητα του Pytorch. Εκτός από την εκπαίδευση του δικτύου δίνει και την δυνατότητα αποθήκευσης του στον δίσκο, μια λεπτομέρεια που δίνει την δυνατότητα να το εκπαιδεύσει και να το δοκιμάσει σε δεύτερο χρόνο εάν αυτός το επιθυμεί. Επίσης μια από τις πιο σημαντικές λειτουργίες που είναι αρκετά ενδιαφέρον για κάποιον που δεν έχει έτοιμα δεδομένα αλλά θέλει να πειραματιστεί γρήγορα και εύκολα, είναι ότι δίνει την δυνατότητα να αποκτήσει κάποιος πολύ γνωστά σύνολα δεδομένων (π χ CIFAR-10, CIFAR-100 και MNIST), αρκεί

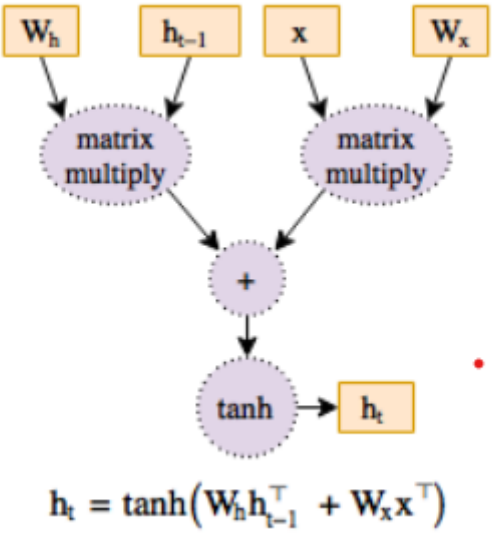
να έχει μια σταθερή πρόσβαση στο διαδίκτυο για να γίνει η ανάκτηση τους απο τις βάσης που διαθέτει το Pytorch.

Γενικά είναι ένα εργαλείο το οποίο δίνει και άλλες πολλές δυνατότητες και είναι αρκετά χρήσιμο για την δημιουργία βαθιών νευρωνικών δικτύων, τα οποία χρησιμοποιούνται κυρίως σε εφαρμογές όπως είναι η αναγνώριση εικόνας. Το περιβάλλον στο οποίο γράφει κάποιος τις εντολές για τον εργαλείο αυτό είναι η γλώσσα προγραμματισμού python η οποία είναι αρκετά εύκολη και χρησιμοποιείται σε έναν μεγάλο βαθμό τα τελευταία χρόνια, από τους ερευνητές και μηχανικούς που ασχολούνται με τον κλάδο του data science. Είναι χρήσιμο να αναφέρουμε ότι δίνει και πολύ καλή υποστήριξη σε GPU μεθόδους, κάτι που δίνει καλύτερη επίδοση κατά την εκπαίδευση και χρήση του νευρωνικού δικτύου.

Επομένως είναι ένα δυνατό εργαλείο το οποίο κάποιος νέος στον χώρο δεν θα έχει μεγάλο πρόβλημα να το διαχειριστεί εφόσον γνωρίζει αρκετά καλά την γλώσσα προγραμματισμού python. Σαν εργαλείο έχει υιοθετηθεί στην χρήση του από τους ερευνητές / μηχανικούς καθώς είναι ένα εργαλείο το οποίο είναι εύκολο στην χρήση αλλά σου δίνει την δυνατότητα να το χρησιμοποιήσεις αμέσως μετά την εγκατάσταση του στο μηχάνημα που εργάζεται κάποιος. Καθώς δίνει δύναμη στην δημιουργία γρήγορων προτύπων και μικρών προγραμμάτων.

Τα κύρια πλεονεκτήματα που δίνει το Pytorch είναι ότι υπάρχει μια πολύ μεγάλη κοινώτητα να το υποστηρίζει δίνοντας πολύ καλές γραπτές τεκμηριώσεις σχετικά με το εργαλείο και με σεμινάρια για τις λειτουργίες του. Όπως είχαμε πει έχει υποστήριξη στην γλώσσα προγραμματισμού python και με καλή διασύνδεση με πολύ γνωστές βιβλιοθήκες όπως NumPy και επειδή είναι στην γλώσσα προγραμματισμού python οι χρήστες είναι σχετικά εύκολο να το μάθουν. Ωστόσο έχει και καλή διασύνδεση με το εφαρμογές που βρίσκονται στο cloud. Υποστηρίζει CPU, GPU και παράλληλη επεξεργασία καθώς και κατανεμημένη εκπαίδευση. Δηλαδή μπορεί να κατανεμηθεί ανάμεσα σε πολλές CPU και GPU πυρήνες και η εκπαίδευση μπορεί να γίνει σε πολλαπλές GPU σε διαφορετικά μηχανήματα. Υπάρχει το Pytorch Hub μια βάση με προ εκπαιδευμένα μοντέλα τα οποία μπορούν να ανακτηθούν, που όπως είπαμε είναι πάνω στην κοινότητα για το ποιά θεωρείτε σημαντικά. Δίνει την δυνατότητα να δημιουργηθούν νέα πράγματα σαν υποκλάσεις της κλασικής κλάσης της python. Ωστόσο παράμετροι μπορούν εύκολα να διαμοιραστούν ανάμεσα σε εξωτερικά εργαλεία όπως το TensorBoard και σε βιβλιοθήκες που μπορούν να εισαχθούν. Τελος έχει μια μεγάλη συλλογή από εργαλεία και βιβλιοθήκες για τομείς που κυμαίνονται από την όραση υπολογιστών έως την ενισχυτική μάθηση.

Το κύριο δομικό στοιχείο του Pytorch είναι τα Tensors και τα Graphs. Τα Tensors είναι κύρια δομικά στοιχεία όμοια με πολυδιάστατους πίνακες και χρησιμοποιούνται για την αποθήκευση και επεξεργασία των εισόδων και εξόδων ενός μοντέλου. Τα Tensors είναι όμοια με τα Numpy ndarrays με την διαφορά ότι τα Tensor μπορούν να χρησιμοποιηθούν σε GPU για επιτάχυνση των υπολογισμών. Απο την άλλη τα Graphs είναι δομές δεδομένων που αποτελείται από συνδεδεμένους κόμβους και ακμές. Τα σύγχρονα εργαλεία βαθιάς μάθησης είναι βασισμένα στην ιδέα των graphs. Το Pytorch κρατάει τα tensors και τις εκτελεσθείσες πράξεις σε κατευθυνόμενους άκυκλους γράφους, όπου τα φύλα είναι οι είσοδοι και οι ρίζες είναι οι έξοδοι των tensors (βλέπε Εικόνα 2).



Εικόνα 2. Απεικόνιση σχήματος γράφων απο πηγή [34]

Το Pytorch είναι βασισμένο σε γραφήματα δυναμικών υπολογισμών όπου ο γράφος έχει δημιουργηθεί και επαναδημιουργείται κατα την εκτέλεση του προγράμματος, με τον ίδιο κώδικα που εκτελεί τους υπολογισμούς για το μπροστινό πέρασμα δημιουργώντας παράλληλα την δομή που είναι απαραίτητη για το backpropagation.

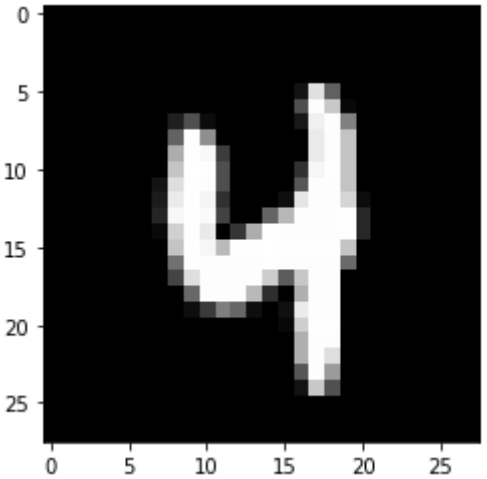
Επίσης το Pytorch μιας και έχει πολύ καλή διασύνδεση με εξωτερικά εργαλεία για κάποιον που θα ήθελε να φτιάξει τα γραφήματα του εκμεταλλευόμενος εργαλεία που έχει το TensorFlow, ένα εργαλείο αρκετά όμοιο με το Pytorch, μπορεί να το διασυνδέσει με το TensorBoard με την βοήθεια εγχειριδίου πού έχει φτιάξει η κοινότητα. Το εγχειρίδιο παρέχει πληροφορίες για τον τρόπο χρήσης με κατανοητό τρόπο.

Απο την ενασχόληση μου με το εργαλείο το βρήκα μια αρκετά καλή επιλογή για να ασχοληθεί κάποιος. Για κάποιον που θέλει να ασχοληθεί γρήγορα με την αναγνώριση εικόνας η δυνατότητα κατεβάσματος του συνόλου δεδομένων από βάση με χρήση μιας μεθόδου είναι ιδιαίτερα χρήσιμο. Η λίστα με τα σύνολα δεδομένων που παρέχει είναι ιδιαίτερα μεγάλη για περισσότερα στο [41]. Από την εγκατάσταση μέχρι και την χρήση δεν υπήρξαν προβλήματα ο κώδικας που έπρεπε να γραφτεί για τις διάφορες λειτουργίες ήταν αρκετά εύκολος και κατανοητός. Ο τρόπος εναλλαγής μηχανημάτων δηλαδή εάν θα το τρέξουμε πάνω σε CPU ή GPU ήταν πολύ εύκολος και για έναν ερευνητή που μπορεί να έχει πολύ λίγους πόρους, όπως το να μην έχει GPU, δίνει πολύ μεγάλη δύναμη. Αν και οι διάφορες επιλογές μπορεί να έχουν επιπτώσεις στην ταχύτητα εκτέλεσης των προγραμμάτων, αυτό δεν σημαίνει ότι δεν μπορούμε να επωφεληθούμε από το εργαλείο. Χωρίς GPU μπορούμε να τρέξουμε αρκετά απλά μοντέλα για να δοκιμάσουμε και να δούμε διάφορες μεθόδους που μπορεί να μας ενδιαφέρουν, με γρήγορο και εύκολο τρόπο. Το γεγονός ότι έχει πολύ καλή διασύνδεση με εξωτερικά εργαλεία και το μέγεθος της κοινότητας δίνει σιγουριά σε όποιων το χρησιμοποιεί ότι δεν υπάρχει κάτι που θα μπορούσε να του σταθεί εμπόδιο στην έρευνα του.

Στα πλαίσια της εργασίας η εναλλαγή μεταξύ των μηχανών (CPU ή GPU) ήταν ιδιαίτερα χρήσιμη καθώς δεν υπήρχε πάντα πρόσβαση σε υπολογιστικό σύστημα με GPU συγκεκριμένων προδιαγραφών που θα μας βόλευε. Επομένως κάθε εκπαίδευση και επαλήθευση του μοντέλου έγινε με την χρήση της CPU του υπολογιστικού συστήματος που ήταν διαθέσιμο. Για πολύπλοκα μοντέλα αυτό ήταν ένα μεγάλο πρόβλημα και για αυτό καταλήξαμε στην χρήση απλών μοντέλων που να μπορεί να χρησιμοποιηθεί εύκολα από όλους.

3.2 Το σύνολο δεδομένων MNIST

Το σύνολο δεδομένων MNIST είναι μια μεγάλη βάση με εικόνες από χειρόγραφους αριθμούς. Είναι ένα πολύ διαδεδομένο και χρησιμοποιημένο σύνολο δεδομένων για εφαρμογές αναγνώρισης εικόνας ή ακόμα και για επεξεργασία εικόνας. Γενικά είναι ένα καλό σύνολο δεδομένων που βοηθάει τους ανθρώπους να δοκιμάσουν τεχνικές μηχανικής μάθησης χωρίς ιδιαίτερη προεπεξεργασία στο σύνολο. Οι εικόνες είναι ασπρόμαυρες και οι αριθμοί κλάσεις που περιέχονται είναι από το 0 έως το 9 και έχουν διαστάσεις 28x28 pixels. Εμπεριέχονται στο σύνολο δεδομένων 70.000 εικόνες, από τις οποίες οι 60.000 εικόνες προορίζονται για την εκπαίδευση των δικτύων ενώ οι υπόλοιπες 10.000 προορίζονται για επαλήθευση. Τα μισά δεδομένα και απο τις δύο λίστες, εκπαίδευσης και επαλήθευσης, προέρχονται από το NIST σύνολο δεδομένων, εκπαίδευσης και επαλήθευσης αντίστοιχα. Όλες οι εικόνες έχουν ετικέτα με τον αριθμό που απεικονίζουν. Παρακάτω φαίνεται η εικόνα ενός αριθμού όπως ανακτήθηκε από το MNIST.



Εικόνα 3. απεικόνιση του χειρόγραφου αριθμού τέσσερα από το σύνολο δεδομένων MNIST

Υπάρχει και ένα νεότερο σετ δεδομένων το Extended MNIST το οποίο θα είναι ο τελικό διάδοχος του MNIST. Το οποίο δεν θα εμπεριέχει μόνο χειρόγραφους αριθμούς αλλα και χειρόγραφα γράμματα κεφαλαία και μικρά. Σαν σύνολο δεδομένων θα είναι στις ίδιες διαστάσεις με το MNIST και μορφή. Σκοπός είναι τα εργαλεία που θα έχουν χρησιμοποιήσει το MNIST θα μπορούν να συνεργαστούν και με το νεότερο Extended MNIST χωρίς ιδιαίτερες εως καθόλου αλλαγές.

Στα πλαίσια της εργασίας θεωρήθηκε ως το καταλληλότερο σύνολο δεδομένων καθώς είναι απλό στην μορφή του και θα επιφέρει πιθανόν καλύτερα οπτικά αποτελέσματα στις τεχνικές, δηλαδή δεν θα είναι τόσο χαοτικά. Έτσι κάθε άνθρωπος που μπορεί να μελετήσει τα αποτελέσματα να είναι εύκολο να τα διαβάσει.

3.2.1 Τρόπος χρήσης μέσα στην εφαρμογή

Ο κώδικας που χρειάζεται για να κατεβασουμε στον υπολογιστη αλλα και να το χρησιμοποιήσουμε τοπικά το σετ δεδομένων είναι ιδιαίτερα απλός μιας και το εργαλείο Pytorch έχει έτοιμη διασύνδεση μέσω μεθόδου, που μας βοηθάει να κάνουμε και τις δύο ενέργειες σε ένα βήμα. Πρακτικά κοιτάει εάν υπάρχει στον κατάλληλο φάκελο, που του δίνουμε ως παράμετρο, κατεβασμένα τα δεδομένα και εάν υπάρχουν τότε προχωράει σε διαδικασία φορτώματος στην μνήμη. Παράλληλα δίνεται και η δυνατότητα να ορίσουμε εάν θέλουμε να εφαρμόσουμε το σετ για εκπαίδευση ή για

επαλήθευση, ακόμη και εάν θέλουμε να τα κατεβάσουμε τα δεδομένα που ζητάμε. Παρακάτω φαίνεται στιγμιότυπο κώδικα που δημιουργήσαμε για την δημιουργία του συνόλου εκπαίδευσης.

```
def get_train_loader(dataset_name, transform, batch_size, num_workers):
    path = f'./data_{dataset_name}';

    trainset = torchvision.datasets.MNIST(root=path, train=True,
                                          download=True, transform=transform);
    trainloader = torch.utils.data.DataLoader(trainset, batch_size=batch_size,
                                              shuffle=True, num_workers=num_workers);
    return trainloader, trainset;
```

Εικόνα 4. Στιγμιότυπο κώδικα κατεβάσματος/χρήσης του σετ εκπαίδευσης

```
def get_test_loader(dataset_name, transform, batch_size, num_workers):
    path = f'./data_{dataset_name}';

    testset = torchvision.datasets.MNIST(root=path, train=False,
                                         download=True, transform=transform)
    testloader = torch.utils.data.DataLoader(testset, batch_size=batch_size,
                                             shuffle=False, num_workers=num_workers);
    return testloader, testset;
```

Εικόνα 5. Στιγμιότυπο κώδικα κατεβάσματος/χρήσης του σετ επαλήθευσης

Παρατηρούμε και για τα δύο στιγμιότυπα ότι δεν έχουν κάποια ιδιαίτερη αλλαγή, μάλιστα θα μπορουσε να είναι και ο ίδιος κώδικας κατάλληλα τροποποιημένος αλλα θεωρείται ότι με την χρήση αυτου του τρόπου δεν θα υπάρχουν προβλήματα με τον κώδικα αργότερα. Μια σημαντική λεπτομέρεια που κρατήσαμε είναι ότι κάθε φορά που θα κατεβάζουμε τα δεδομένα, τα δεδομένα επαλήθευσης δεν θα τα διαβαζουμε τυχαία αλλά θα έρχονται με την ίδια σειρα. Αυτό γίνεται για να μας βοηθήσει αργότερα να κρατάμε ίδιες συμπεριφορές στις μεθόδους μας και να μπορούμε να τα συγκρίνουμε μεταξύ τους λίγο πιο εύκολα εάν χρειαστεί. Οι πληροφορίες αυτές σχετικά με το πώς χρησιμοποιείς το σύνολο δεδομένων με την βοήθεια του Pytorch μπορεί να βρεθεί μέσω της πηγής [39]. Το σημαντικό κομμάτι σε αυτήν την μέθοδο είναι το testloader/trainloader, ανάλογα την περίπτωση, μας βοηθάει για να διαβάσουμε τα δεδομένα σε δισδιάστατη λίστα μεγέθους ανάλογη της παραμέτρου batch_size που περνάμε κάθε φορά.

Για να διαβαστούν κατάλληλα οι εικόνες κάθε batch πρέπει να αρχικοποιηθεί ένας iterator πάνω στον οποίο εκτελώντας την κατάλληλη μέθοδο next πάνω του θα μας δώσει ένα tuple που θα περιέχει την λίστα με τις εικόνες και την λίστα με τις ετικέτες κάθε εικόνας, μια προς μία αντιστοίχηση. Σημαντικό είναι να αναφέρουμε ότι οι εικόνες έχουν υποστεί τις τροποποιήσεις που έχουμε εισάγει απο τον transformer που χρειάζεται για να δουλέψουμε με το Pytorch. Η κύρια τροποποίηση που δέχονται είναι να τροποποιηθούν ώστε να γίνουν Tensors.

```
def get_images_labels(image_loader):
    dataiter = iter(image_loader);
    images, labels = next(dataiter);

    return images, labels;
```

Εικόνα 6. Στιγμιότυπο κώδικα που διαβάζει τις εικόνες από τον image_loader

Αφου πάρουμε αυτά τα δεδομένα θα χρειαστεί να τα μετατρέψουμε σε nparray ώστε να μπορέσουμε να τα δείξουμε στην οθόνη μέσω τον plots. Όλη η μεθοδος θα πρέπει να γίνεται επαναληπτικά για κάθε batch εάν είναι επιθυμητό. Η παρακάτω μέθοδος έχει τροποποιηθεί έτσι ώστε να δείξει μια ολόκληρη λίστα με εικόνες σε ένα plot αλλά χρησιμοποιείται συνήθως για ένα πολύ μικρό πλήθος από δεδομένα, από μία εικόνα μέχρι και έντεκα που συνήθως θα χρειαζόμασταν να δείξουμε, μια για κάθε κλάση (δέκα κλάσεις έχει το σετ δεδομένων).

```
def show_tensor_image(images):
    image = torchvision.utils.make_grid(images)
    image = image.numpy()
    plt.imshow(np.transpose(image, (1, 2, 0)))
    plt.show()
```

Εικόνα 7. Στιγμιότυπο κώδικα που μετατρέπει τις εικόνες στην κατάλληλη μορφή για να τα δείξει.

3.3 Ανάλυση τεχνικών

Οι τεχνικές που επιλέχθηκαν απο τις πιο γνωστές τεχνικές επεξηγήσιμης τεχνητής νοημοσύνης είναι το LIME και το Deep SHAP. Δύο τεχνικές που φαίνεται να έχουν αναφερθεί αρκετά σε διάφορες εφαρμογές όπως είναι η αναγνώριση εικόνας και αναγνώριση αντικειμένου. Η επιλογή των μεθόδων έγινε λόγω της μεγάλης συζήτησης γύρω από αυτές από την κοινότητα και την διαθεσιμότητα βιβλιοθηκών που τις υλοποιούν.

Παρακάτω θα αναλύσουμε περισσότερο το θεωρητικό μέρος κάθε μεθόδου που επιλέχθηκε καθώς και τον τρόπο χρήσης μέσα στην εφαρμογή, βάση των διάφορων οδηγών χρήσης από τους οποίους αντλήσαμε την κατάλληλη πληροφορία που μας ενδιαφέρει. Γενικά ο τρόπος χρήσης κάθε μεθόδου δεν ήταν κάτι δύσκολο αλλά κάθε τεχνική έχει και τις ιδιαιτερότητες της κατα την χρήση, λόγω της υλοποιήσεις που έχει γίνει για κάθε τεχνική. Περισσότερες πληροφορίες σχετικά με την χρήση θα αναφερθεί και στην ενότητα 3.4 Προεπεξεργασία.

3.3.1 LIME

Το LIME (Local Interpretable Model-agnostic Explanations) για να καταφέρει να εξηγήσει μια είσοδο χρησιμοποιεί ένα τοπικό γραμμικό μοντέλο ώστε να προσεγγίσει την συμπεριφορά του αδιαφανούς μοντέλου. Μπορεί να χρησιμοποιηθεί για μοντέλα που ταξινομούν δεδομένα σε πίνακα, εικόνες ή κείμενα. Με την έννοια του Interpretable στην ονομασία εννοεί ότι βοηθά να κατανοήσουμε γιατι ένα μοντέλο συμπεριφέρεται με τον τρόπο που συμπεριφέρεται.

Η τεχνική αυτή αποτελείται από τέσσερα βασικά βήματα. Στο πρώτο βήμα το LIME δημιουργεί πολλές διαταραγμένες εικόνες ενεργοποιώντας και απενεργοποιώντας ορισμένα απο τα super-pixel της εικόνας. Σε ένα δεύτερο βήμα, μια κλάση πρόβλεψης, για κάθε τεχνητά δημιουργημένη εικόνα, δημιουργείται χρησιμοποιώντας το εκπαιδευμένο μοντέλο. Σε ένα τρίτο βήμα, υπολογίζονται βάρη για κάθε τεχνητή εικόνα για να υπολογιστεί το μέτρο σημαντικότητας. Η απόσταση υπολογίζεται ανάμεσα σε κάθε σημείου εικόνας που δημιουργείται τεχνητά και των αντίστοιχων σημείων της αρχικής εικόνας εισόδου. Η απόσταση αντιστοιχίζεται ως βάρος με τιμές που κυμαίνονται από 0 έως 1. Η πιο κοντινή εγγύτητα του διαταραγμένη στιγμιότυπου, με το παράδειγμα που εξηγείται, συμβάλει στο υψηλότερο σχετικό βάρος. Και το τελευταίο βήμα περιλαμβάνει την προσαρμογή ενός γραμμικού μοντέλου με την βοήθεια των τεχνητά σταθμισμένων σημείων. Με αυτό τον τρόπο ο προσαρμοσμένος συντελεστής λαμβάνεται για κάθε χαρακτηριστικό. Τα super-pixel που αντιστοιχούν σε υψηλότερες τιμές συντελεστών είναι αυτά που συμβάλλουν σημαντικά στην πρόβλεψη του μοντέλου μηχανικής μάθησης.

Τα σημαντικότερα χαρακτηριστικά οπτικοποιούνται ώς περιοχές, πάνω στην εικόνα εισόδου, που μας δείχνουν τα σημαντικότερα χαρακτηριστικά που επηρεάζουν τις προβλέψεις του μοντέλου. Σαν τεχνική χρειάζεται πολλά περάσματα μέσω του νευρωνικού δικτύου, κάτι που θεωρείται ώς πολύ ακριβή διαδικασία συγκριτικά με άλλες μεθόδους επεξήγησης προβλέψεων.

Η εξήγηση της μεθόδου μπορεί να υπολογιστεί απο την εξής εξίσωση:

$$y(\chi) = argmin\, L\left(f, g, \pi_{\chi}\right) + \Omega(g) \qquad (1)$$

Όπου για ένα στιγμιότυπο x, η εξήγηση ορίζεται ώς το μοντέλο $g \in G$, οπου G είναι η κλάση πιθανόν επεξηγήσημων μοντέλων. Το $\Omega(g)$ είναι η πολυπλοκότητα του μοντέλου g, στοχεύοντας να αποκλείσει τα πολύπλοκα μοντέλα από την ερμηνεύσιμη κατηγορία G. Το $L\left(f, g, \pi_{\chi}\right)$ υπολογίζει πόσο αναξιόπιστο είναι το g προσεγγίζοντας το f στην τοποθεσία γύρω από το x που ορίζεται από το π_{χ}. Το $f: R^p \rightarrow R$ είναι η προβλεπόμενη πιθανότητα να ανήκει σε μία κλάση και το π_{χ} είναι το μέτρο εγγύτητας ανάμεσα σε στιγμιότυπα z και x. Ως g επιλέγεται ανάλογα και απο τα δύο το L και Ω προσπαθεί να ελαχιστοποιήσει πόσο κοντά είναι η εξήγηση στην πρόβλεψη του αρχικού μοντέλου f όσο η πολυπλοκότητα του μοντέλου $\Omega(g)$ παραμένη χαμηλά.

Μέχρι εδώ έχουν αναφερθεί αρκετά σημαντικές θεωρητικές πληροφορίες που βοηθούν στην καλύτερη κατανόηση της μεθόδου και του τι περιμένουμε να δούμε στο κεφάλαιο των αποτελεσμάτων αργότερα. Παρακάτω θα αναφερθεί ο τρόπος χρήσης της μεθόδου κατα την υλοποίηση της εφαρμογής καθώς και θα αναφερθεί ο ορισμός των παραμέτρων της μεθόδου μέσα από τον οδηγό χρήσης.

Για να καταφέρουμε να χρησιμοποιηθεί η μέθοδος LIME με την χρήση κώδικα χρειάζεται να περάσουμε ως παραμέτρους την εικόνα που θέλουμε να γίνει η εξήγηση, η εικόνα που ζητάει να περαστεί κάθε φορά θα πρέπει να είναι σε RGB μορφοποίηση. Η δεύτερη παράμετρος που χρειάζεται είναι η μέθοδος πρόβλεψης για να μπορέσει το LIME να λαμβάνει τις πρεντύξεις του μοντέλου που χρησιμοποιούμε όταν αυτό χρειαστεί. Η συνάρτηση πρόβλεψης με την σειρα της λαμβάνει ως παράμετρο την εικόνα που περάσαμε στην κύρια μέθοδο με τις κατάλληλες αλλαγές που επιβάλλει ο αλγόριθμος του LIME κάθε φορά. Μια σημαντική παράμετρος που έχει η μέθοδος του LIME είναι ο αριθμός των ετικετων που θέλουμε. Ωστόσο μπορούμε να διαλέξουμε και τον αριθμό των δειγμάτων που λαμβάνει η μέθοδος. Τελευταία και σημαντική παράμετρος είναι η συνάρτηση κατάτμησης, η μέθοδος LIME χρησιμοποιεί τον αλγόριθμο quickshift. Στον αλγόριθμο κατάτμησης

παραμετροποιήσαμε την τιμή του kernel_size, αρχική τιμή είναι τέσσερα αλλά εμείς το θέσαμε ως ένα για να έχουμε πιο λεπτομερείς εξηγήσεις, μεγαλύτερος αριθμός οδηγεί σε λιγότερες συστάδες και λιγότερο λεπτομερείς εξηγήσεις. Παρακάτω φαίνεται η παραμετροποίηση που χρησιμοποιήθηκε και ο κώδικας που εκτελεί την κύρια δουλειά της επεξήγησης.

```
def explain_with_lime(image, label, predict_fn):
    segmenter = SegmentationAlgorithm('quickshift', kernel_size=1, max_dist=200, ratio=0.2)
    explainer = lime_image.LimeImageExplainer(verbose=False)
    explanation = explainer.explain_instance(image,
                                             predict_fn,
                                             top_labels=10,
                                             hide_color=0,
                                             num_samples=10000,
                                             segmentation_fn=segmenter)

    # now show them for each class
    fig, m_axs = plt.subplots(2,5, figsize = (12,6))
    for i, c_ax in enumerate(m_axs.flatten()):
        temp, mask = explanation.get_image_and_mask(i, positive_only=True,
                                                    num_features=10,
                                                    hide_rest=False,
                                                    min_weight = 0.01 )
        c_ax.imshow(label2rgb(mask,image, bg_label = 0), interpolation = 'nearest')
        c_ax.set_title('Positive for {}\nActual {}'.format(i, label))
        c_ax.axis('off')
    plt.show()
```

Εικόνα 8. Κώδικας επεξήγησης με την χρήση μεθόδου LIME

Παρατηρούμε παραπάνω στην Εικόνα 6 ότι η συνάρτηση που κάνει όλη την δουλειά είναι η μέθοδος explain_instance και εμπεριέχει όλες τις απαραίτητες παραμέτρους που περιγράφηκαν. Οι υπόλοιπες γραμμές κώδικα είναι απλά για να καταφέρουμε να δείξουμε τα αποτελέσματα σε κατάλληλα plots. Η μεταβλητή explanation που φαίνεται στην εικόνα εμπεριέχει όλες τις κατάλληλες πληροφορίες και μεθόδους που μπορεί να χρειαστεί κάποιος κατα την δοκιμή του με την μέθοδο LIME, στα πλαίσια της εργασίας η σημαντική μέθοδος που χρειάστηκε είναι η get_image_and_mask όπου χρειάζεται να της δώσουμε ως πληροφορία την ετικέτα που ψάχνουμε, εάν θέλουμε να εμφανίσουμε την μάσκα μόνο όσων επηρεάζουν για την πρόβλεψη του μοντέλου, με την τρίτη παράμετρο επηρεάζεται ο αριθμός των super-pixel που θέλουμε να λάβει υπόψη, με την παράμετρο hide_rest εάν έχει λάβει τιμή True τότε εμφανίζει στην εικόνα, που μας επιστρέφει, τις περιοχές που δεν λαμβάνονται υπόψη από το μοντέλο ως γκρι περιοχή αυτή η επιλογή ίσως είναι πιο βοηθητική σε ποιο πολύπλοκες εικόνες και τέλος ορίζεται το χαμηλότερο βάρος που χρειάζεται να έχουν τα super-pixel ώστε να συμπεριληφθούν στην επεξήγηση.

3.3.2 SHAP

Η κλασσική τεχνική SHAP θεωρείται ως αγνωστικιστική μέθοδος (model-agnostic) η οποία μπορεί να χρησιμοποιηθεί ώστε να εξηγήσει ατομικές απαντήσεις από οποιοδήποτε μοντέλο μηχανικής μάθησης. Είναι βασισμένη σε μια προσέγγιση της θεωρίας παιγνίων για τη συσσώρευση πρόσθετης συνεισφοράς όλων των χαρακτηριστικών που εμπλέκονται σε ένα μοντέλο. Η μέθοδος αναθέτει σε κάθε χαρακτηριστικό μια τιμή σημαντικότητας μέσα σε ένα σύνολο προσδοκιών υπό όρους για μια συγκεκριμένη πρόβλεψη. Τα αποτελέσματα αυτής της διαδικασίας ονομάζονται τιμές SHAP (shapley values). Αυτές οι τιμές μπορούν να κατανεμηθούν σε μια βασική τιμή, έπειτα η μέθοδος θα απαριθμήσει γραφικά την συνεισφορά όλων των χαρακτηριστικών στην τιμή SHAP

προσδιορίζοντας ποια χαρακτηριστικά συνέβαλαν στο να είναι μεγαλύτερη ή μικρότερη από τη βασική τιμή. Επειδή η μέθοδος SHAP λαμβάνει υπόψη όλα τα χαρακτηριστικά, η μέθοδος μπορεί να είναι αρκετά ακριβή υπολογιστικά σε μεγάλα μοντέλα, όμως είναι απαραίτητη για την κατανόηση μοντέλων μηχανικής μάθησης.

Με μαθηματικό υπόβαθρο η τιμή SHAP ορίζεται ως εξής. Η μέθοδος λαμβάνει υπόψη κάθε υποσύνολο συνεργαζόμενων τιμών $S \subseteq P$, όπου P σύνολο συνεργαζόμενων τιμών. Δοκιμάζει πως επηρεάζει μια τιμή i στο S σχετικά με την συνολική απόδοση $u(S)$ που λαμβάνεται από το S εάν συνεργαστούν. Οι τιμές SHAP προσδιορίζουν τη συνεισφορά της τιμης i στο συνολικό P με την εξίσωση:

$$\varphi_i = \sum_{S \subseteq P \setminus \{i\}} as \cdot \left[u\left(S \cup \{i\}\right) - u(S) \right] \qquad (2)$$

Όπου κάθε υποσύνολο S είναι σταθμισμένο απο το παράγοντα $as = \frac{|S|! \cdot (|P| - |S| - 1)!}{|P|!}$. Οι τιμές SHAP ικανοποιούν τα αξιώματα, αποδοτικότητα ($\sum \varphi_i = u(P)$), συμμετρία, γραμμικότητα και μηδενική προστιθέμενη αξία μιας εικονικής τιμής.

Για την εξήγηση των βαθιών νευρωνικών δικτύων κυρίος χρησιμοποιείται ο αλγόριθμος Deep lift για να προσεγγίσει τις τιμές SHAP (Deep SHAP). Η τεχνική αυτή είναι μη αγνωστικιστική (model-specific) και δίνει τις εξηγήσεις βασιζόμενη στην διαφορά ανάμεσα στις εξόδους κάποιων παραδειγμάτων και στην διαφορά με την είσοδο που δέχεται να εξηγήσει.

Μέχρι εδώ έχουν αναφερθεί αρκετά σημαντικές θεωρητικές πληροφορίες που βοηθούν στην καλύτερη κατανόηση της μεθόδου και του τι περιμένουμε να δούμε στο κεφάλαιο των αποτελεσμάτων αργότερα. Παρακάτω θα αναφερθεί ο τρόπος χρήσης της μεθόδου κατα την υλοποίηση της εφαρμογής καθώς και θα αναφερθεί ο ορισμός των παραμέτρων της μεθόδου μέσα από τον οδηγό χρήσης.

Για να καταφέρουμε να χρησιμοποιήσουμε την τεχνική SHAP σε συνδυασμό με τον αλγόριθμο DeepLIFT δίνεται η δυνατότητα του Deep Explainer. Για την χρήση του με κώδικα χρειάζεται να περάσουμε στον Deep Explainer το μοντέλο μας και τα παραδείγματα υποβάθρου που απαιτεί. Αφού τα δώσουμε αυτά μας επιστρέφει έναν explainer όπου με την κατάλληλη εντολή υπολογίζει τα SHAP values για τις εικόνες που επιθυμουμε να δώσουμε μια εξήγηση, χρησιμοποιώντας τα SHAP values και συνδυάζοντας τα με την εικόνα μας μπορούμε να τυπώσουμε στην οθόνη τα απαραίτητα plot.

```
def explain_with_shap(model, background, test_images , labels):
    explainer = shap.DeepExplainer(model, background)
    shap_values = explainer.shap_values(test_images)
    shap_numpy = [np.swapaxes(np.swapaxes(s, 1, -1), 1, 2) for s in shap_values]
    test_numpy = np.swapaxes(np.swapaxes(test_images.numpy(), 1, -1), 1, 2)
    # plot the feature attributions
    shap.image_plot(shap_numpy, -test_numpy, labels= labels)
```

Εικόνα 9. Μέθοδος εξήγησης με Deep Shap

Όπως παρατηρούμε και παραπάνω η χρήση με κώδικα είναι αρκετά απλή και εύκολη διαδικασία το μόνο που θα χρειαστει να επεξεργαστούμε αργότερα είναι η μεταβλητή background που περιέχει όλα

τα παραδείγματα που χρειάζεται για να συγκρίνει με τις εικόνες που θέλουμε να λάβουμε τις κατάλληλες εξηγήσεις. Ωστόσο η shap μεταβλητή που φαίνεται στην *Εικόνα 8* είναι η κλάση που περιέχει όλες τις απαραίτητες μεθόδους που μπορεί να χρησιμοποιήσει κάποιος μηχανικός ανάλογα τις ανάγκες του και τους αλγορίθμους με τους οποίους θέλει να πειραματιστεί.

3.4 Προεπεξεργασία

Η προεπεξεργασία που χρειάστηκε να γίνει ήταν ανάλογη του στόχου που χρειάστηκε να επιτευχθεί ανα μέθοδο. Δηλαδή υπήρξε διαφοροποίηση ανάμεσα στην βασική επεξεργασία των δεδομένων που χρησιμοποιήθηκε από την εκπαίδευση του μοντέλου μέχρι και κάθε μέθοδο επεξήγησης. Η βασική προεπεξεργασία που γίνεται είναι να μετατρέπει τα δεδομένα του συνόλου δεδομένων MNIST στην βασική μορφή Tensor που χρειάζεται να διαχειριστεί το εργαλείο Pytorch, όπως είχε αναφερθεί και στην ενότητα "3.1 Λίγα λόγια για το Pytorch" και έγινε κανονικοποίηση του συνόλου δεδομένων με mean (ακολουθία μέσων για κάθε κανάλι) στο 0.5 και standard (ακολουθία τυπικών αποκλίσεων για κάθε κανάλι.) στο 0.5 απόκλιση και εφαρμόζεται πάντα μορφοποίηση σε Pil Image μια βιβλιοθήκη της python που βοηθά στην επεξεργασία των εικόνων, τέλος εφαρμόζεται πάλι η μορφοποίηση σε Tensor που μπορεί να διαχειριστεί το εργαλείο Pytorch.

```
def get_transform():
    return transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize(mean=(0.5,), std=(0.5,)),
        transforms.ToPILImage(),
        transforms.ToTensor()
    ]);
```

Εικόνα 10. Βασική προεπεξεργασία

Για να χρησιμοποιηθεί κατάλληλα το LIME η παραπάνω επεξεργασία δεν ήταν αρκετή. Το LIME διαχειρίζεται εικόνες σε μορφοποίηση RGB, ενώ το σύνολο δεδομένων που χρησιμοποιείται στην περίπτωση της εργασίας είναι σε ασπρόμαυρη μορφοποίηση και έχουν εφαρμοστεί και οι βασικές μετατροπές. Οπότε για να χρησιμοποιηθεί το LIME πρέπει πρώτα να περάσει η εικόνα απο μια μορφοποίηση ώστε να γίνει κανονική εικόνα και με την χρήση της νέας μορφοποίησης να την μετατρεψουμε σε RGB, δυστυχώς η αρχική μορφοποίηση δεν μπορεί να μην γίνει μιας και η κανονικοποίηση (normalize) μπορεί να εφαρμοστεί μόνο σε Tensor μορφοποίηση. Υπάρχει και η δυνατότητα απο την υλοποίηση του LIME να περαστεί ασπρόμαυρη εικόνα αλλά μέσα θα την μετατρέψει σε RGB, οπότε θεωρήθηκε ότι πρέπει να το ελέγχουμε εμείς όσο γίνεται και να μην κάνει το εργαλείο ότι πιστεύει εφόσον αυτό είναι εφικτό. Όλη αυτή η αλλαγή της εικόνας προσθέτει παραπάνω λογική στην συνάρτηση πρόβλεψης που αναφέραμε στην "3.3 Ανάλυση τεχνικών", είναι η συνάρτηση που αλληλεπιδρά το μοντέλο μας με την εικόνα που περάσαμε. Σε αυτή την συνάρτηση θα χρειαστεί να μετατρέψουμε την εικόνα σε ασπρόμαυρη μορφοποίηση και πάλι σε Tensor ώστε να χρησιμοποιηθεί από το μοντέλο. Μπορεί να φαίνεται περιπλοκο αλλά είναι η πιο απλή λύση που μπορεί να χρησιμοποιήσει κάποιος εάν το μοντέλο του διαχειρίζεται ασπρόμαυρες εικόνες, ούτως η άλλως το LIME από μόνο κάνει παρόμοια μετατροπή επομένως δεν χρειάζεται να ανησυχούμε τόσο πολύ. Μια άλλη λύση που θα μπορούσε να χρησιμοποιήσει κάποιος είναι να αλλάξει το μοντέλο του σε ένα μοντέλο που μπορεί να αναγνωρίσει RGB εικόνες έναντι των ασπρόμαυρων, αυτό όμως δεν

είναι εφικτό πάντα σε όλες τις εφαρμογές ειδικά των εφαρμογών που έχουν υλοποιηθεί πριν από μεγάλο χρονικό διάστημα.

```
rgb_image = hlp.tensor_to_RGB_image(image);
explain_with_lime(rgb_image, image_label, batch_predict,
                  predicted_label);
```

Εικόνα 11. Μετατροπή της εικόνας σε RGB.

```
def batch_predict(images):
    batch = torch.stack(tuple(
        hlp.nparray_to_gray_scale_image(i, hlp.get_transform_Tensor())
                                for i in images), dim=0
        )
    batch.to(device);
    return hlp.get_probs(model, batch);
```

Εικόνα 12. Η συνάρτηση πρόβλεψης

Όσο για τον χωρισμό των δεδομένων στην μέθοδο LIME αποφασίστηκε να χρησιμοποιηθεί ο ίδιος αλγόριθμος που χρησιμοποιήθηκε στην μέθοδο SHAP ώστε να έχουν κοινό σημείο αναφοράς περισσότερα για την μεθοδολογία χωρισμου δεδομένων θα αναφερθεί παρακάτω.

Για την περίπτωση της μεθόδου Deep SHAP η επεξεργασία που χρειάζεται να γίνει είναι πιο απλή. Όπως έχει αναφερθεί η βασική διαδικασία με την μετατροπή του συνόλου δεδομένων δεν μπορεί να λείψει, η υλοποίηση που χρησιμοποιήθηκε μπορεί να διαχειριστεί τα Tensors όπως ακριβώς είναι χωρίς να κάνουμε κάτι παραπάνω. Όμως το Deep SHAP χρειάζεται παραδείγματα υποβάθρου ώστε να συγκρίνει και να υπολογίσει την εξήγηση. Η λίστα που επιλέχθηκε να εισαχθεί κάθε φορά, περιέχει εκατό (100) εικόνες ανά κλάση, συνολικά μια λίστα με χίλια στοιχεία (1000), από όλες αυτές τις εικόνες δεν περιέχεται εικόνα που δεν μπορεί να αναγνωριστεί σωστά από το μοντέλο, αυτό έγινε για να οριστεί κατάλληλα το σύνολο όπου το μοντέλο πρέπει να μπορεί να διαχειριστεί τις καταστάσεις. Τα παραδείγματα ανά κλάση θεωρήθηκαν αρκετά λόγω διαφορετικότητας. Στην μέθοδο LIME επειδή δεν υφίσταται λίστα παραδειγμάτων υποβάθρου μπορει να εισαχθεί ως μηδέν (0) έτσι δεν γεμίζεται η λίστα υποβάθρου, σημαντικό είναι να αναφέρουμε ότι ο αλγόριθμος δίνει προτεραιότητα στις λίστες δεδομένων και όχι στην λίστα υποβάθρου.

```
shap_data = split_data(model,
                       datasetmanager.testloader,
                       device,
                       datasetmanager,
                       passed_images_per_class=10,
                       failed_images_per_class=10,
                       background_per_class=100
                       );

background, passed_images_labels, failed_images_labels = shap_data;

test_images, test_labels = passed_images_labels;
test_images = test_images[0:1];
test_labels = test_labels[0:1];

hlp.print_ground_truth(test_images, test_labels, datasetmanager);

explain_with_shap(model, test_images, background,
                  ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']);
```

Εικόνα 13. Επεξεργασία δεδομένων και χρήση μεθόδου Deep SHAP.

Παρατηρούμε ότι στην Εικόνα 13. η μεθοδολογία που πρέπει να ακολουθηθεί είναι ιδιαίτερα απλή συγκριτικά με το LIME. Πιθανότατα αυτό συμβαίνει για δύο βασικούς λόγους, ο ένας λόγος είναι ότι το LIME έχει συγκεκριμένη διαχείριση στην εικόνα εισόδου ανάλογα την μορφοποίηση της καθώς προσπαθεί να είναι όσο αγνωστικιστική μέθοδος γίνεται, αυτό ρίχνει όλη την ευθύνη σε όποιον την χρησιμοποιεί. Ενώ το Deep SHAP είναι μία μέθοδος που είναι στηριγμένη στο ίδιο το μοντέλο, έτσι στην υλοποίηση της μπορούν να λάβουν υπόψη διάφορους παραμέτρους, λίγο πιο εύκολα.

3.5 Το Μοντέλο

Για να μπορέσει να γίνει η επεξήγηση ενός μοντέλου πρέπει να το επιλέξουμε και να δοκιμάσουμε τις μεθόδους πάνω σε αυτό. Έγινε μια προσπάθεια να βρεθεί ένα προ-εκπαιδευμένο μοντέλο με την βοήθεια του εργαλείου Pytorch, θα ήταν ιδανική περίπτωση, αλλα δεν υπάρχει κάτι που να είναι τόσο βοηθητικο για το σύνολο δεδομένων MNIST. Όσα εγχειρίδια βρεθηκαν κατέβαζαν προ-εκπαιδευμένα μοντέλα (άγνωστο σύνολο εκπαίδευσης) και μετά τα επηρέασαν με κώδικα, αυτή η μέθοδος θεωρήθηκε ως μια κακή τακτική και δεν χρησιμοποιήθηκε για τον πειραματισμό. Επομένως για τον πειραματισμό χρειάστηκε να δημιουργηθεί ένα μοντέλο από την αρχή με την βοήθεια του Pytorch. Η μεθοδολογία για την δημιουργία ενός μοντέλου ακολουθήθηκε βάση παραδειγμάτων με το αντίστοιχο σύνολο δεδομένων που χρησιμοποιούμε και φαίνεται να έχουν καλά αποτελέσματα, τουλάχιστον για το σύνολο δεδομένων που έχουμε.

Το μοντέλο το οποίο χρησιμοποιήθηκε περιέχει 2-D συνελικτικά στρώματα ακολουθούμενα από δύο πλήρως συνδεδεμένα στρώματα (κρυφά στρώματα). Σαν λειτουργία ενεργοποίησης επιλέχθηκαν διορθωμένες γραμμικές μονάδες (Relu) και ως μέσο τακτοποίησης χρησιμοποιούνται δύο επίπεδα εγκατάλειψης (dropout). Με το Pytorch όπως είχαμε αναφέρει μπορούμε να δημιουργήσουμε μοντέλα με την βοήθεια κλάσεων. Παρακάτω φαίνεται ο κώδικας που δημιουργήθηκε για το μοντέλο.

```
class Cnn(nn.Module):
    def __init__(self):
        super(Cnn, self).__init__()

        self.conv_layers = nn.Sequential(
            nn.Conv2d(1, 10, kernel_size=5),
            nn.MaxPool2d(2),
            nn.ReLU(),
            nn.Conv2d(10, 20, kernel_size=5),
            nn.Dropout(),
            nn.MaxPool2d(2),
            nn.ReLU(),
        )
        self.fc_layers = nn.Sequential(
            nn.Linear(320, 50),
            nn.ReLU(),
            nn.Dropout(),
            nn.Linear(50, 10)
        )

    def forward(self, x):
        x = self.conv_layers(x)
        x = x.view(-1, 320)
        x = self.fc_layers(x)
        return F.log_softmax(x)
```

Εικόνα 14. Νευρωνικό δίκτυο με την βοήθεια του Pytorch

Στην Εικόνα 14 παρατηρούμε ότι στην μέθοδο __init__ ορίζονται τα κατάλληλα επίπεδα του μοντέλου τα οποία θα χρειαστούμε ως ορισμό αλλα και στην μέθοδο forward που τα χρησιμοποιούμε. Η μέθοδος forward ορίζει τον τρόπο με τον οποίο υπολογίζεται η έξοδος του δικτύου χρησιμοποιώντας τα επίπεδα που ορίσαμε στην __init__.

Για να μπορέσει το μοντέλο να εκπαιδευτεί και να χρησιμοποιηθεί χρειάζεται να οριστεί ένας βελτιστοποιητής (optimizer) και μια λειτουργία απώλειας (loss function). Ο βελτιστοποιητής που χρησιμοποιήθηκε είναι ο SGD (stochastic gradient descent) και λαμβάνει ως παραμέτρους τις παραμέτρους του δικτύου, τον ρυθμό μάθησης όπου και η επιλογή τιμής επιλέχθηκε lr=0.01 και ορμή (momentum), η τιμή που επιλέχθηκε για την ορμή είναι momentum=0.5. Από την άλλη η συνάρτηση απώλειας που χρησιμοποιήθηκε είναι η μέθοδος nll_loss (negative log likelihood loss) όπου σαν παραμέτρους που λαμβάνει όταν αυτό χρησιμοποιείται είναι η έξοδος του μοντέλου και τους κατάλληλους στόχους που πρέπει να ταιριάζει.

```
def get_optimizer(model):
    return optim.SGD(model.parameters(), lr=0.01, momentum=0.5);
```

Εικόνα 15. Κώδικας που δημιουργεί τον βελτιστοποιητή.

```
def get_loss_function():

    def loss_function(output, target):
        return F.nll_loss(output, target);

    return loss_function;
```

Εικόνα 16. Κώδικας που χρησιμοποιήθηκε για χειρισμό της λειτουργίας απώλειας

Παρακάτω θα αναφερθεί η διαδικασία εκπαίδευσης που ακολουθήθηκε καθώς και η διαδικασία επαλήθευσης του μοντέλου. Κάθε μορφή κώδικα που δημιουργήθηκε είτε για το μοντέλο αλλα και για τις διαδικασίες εκπαίδευσης / επαλήθευσης, δημιουργήθηκε με την βοήθεια εγχειριδίων χρήσης του Pytorch.

3.5.1 Διαδικασία εκπαίδευσης

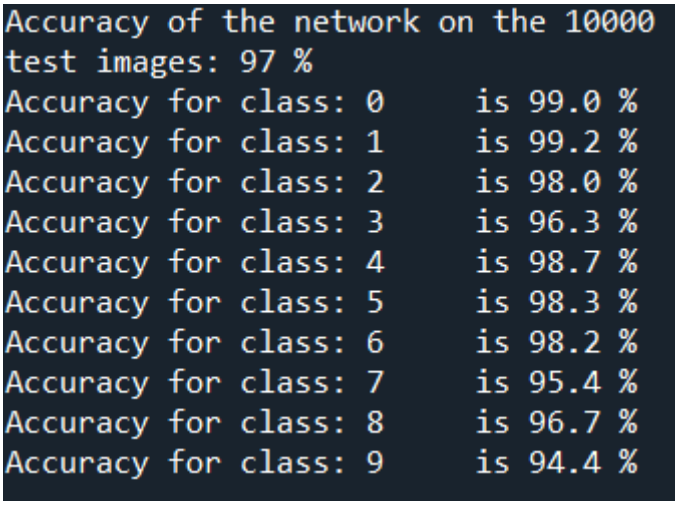
Η διαδικασία εκπαίδευσης περιλαμβάνει τον τρόπο με τον οποίο έγινε η εκπαίδευση και τον τρόπο με τον οποίο μετρήθηκε η απόδοση του δικτύου. Ο αλγόριθμος της εκπαίδευσης δημιουργήθηκε με την βοήθεια του εγχειριδίου χρήσης του Pytorch. Σαν διαδικασία όπως περιγράφεται από το εγχειρίδιο είναι ιδιαίτερα απλή, γίνεται επανάληψη πάνω στο πίνακα δεδομένων, ο οποίος είναι χωρισμένος σε παρτίδες των 128 ώστε να αυξηθεί η ταχύτητα εκτέλεσης, και δίνονται στο μοντέλο ώστε να εκπαιδευτεί, φυσικα και με την βοήθεια του βελτιστοποιητή και της λειτουργίας απώλειας. Όλη αυτή η διαδικασία επαναλαμβάνεται σε εποχές (epochs) τόσες όσες πιστεύουμε ότι θα φτάσουμε σε ένα σχετικά καλό επίπεδο. Μέσα από τα διάφορα παραδείγματα που κυκλοφορούν ο αριθμός επαναλήψεων εκπαίδευσης (epochs) κυμαίνεται από τρεις (3) μέχρι και δέκα (10), στην περίπτωση της εργασίας χρησιμοποιήθηκε να γίνει η διαδικασία δέκα (10) φορές για να δούμε πως θα επηρεαστεί το δίκτυο από την απόφαση αυτή και εάν επιφέρει καλύτερη απόδοση μετα απο αρκετές επαναλήψεις. Σε κάθε περίπτωση ο αλγόριθμος σε μορφή κώδικα φαίνεται παρακάτω ώστε να μπορεί να κατανοηθεί καλύτερα.

```python
def train_classifier(model, trainloader,
                     device, loss_function,
                     optimizer, num_epoch,
                     log_every):
    model.train();
    for epoch in range(num_epoch):  # loop over the dataset multiple times
        print(f'Loop: {epoch+1} out of {num_epoch}')
        running_loss = 0.0
        for index, (inputs, labels) in enumerate(trainloader, 0):

            inputs, labels = inputs.to(device), labels.to(device);

            # zero the parameter gradients
            optimizer.zero_grad()

            # forward + backward + optimize
            outputs = model(inputs);

            loss = loss_function(outputs, labels)
            loss.backward()

            optimizer.step()

            # print statistics
            running_loss += loss.item()
            if (index+1) % log_every == 0:
                print(f'[{epoch + 1}, {index+1:5d}] loss: {running_loss / 2000:.3f}')
                running_loss = 0.0
```

Εικόνα 17. Αλγόριθμος εκπαίδευσης μοντέλου.

3.5.2 Επαλήθευση μοντέλου

Η επαλήθευση του μοντέλου τυπικά γίνεται άμεσα μετά την εκπαίδευση. Η ίδια η διαδικασία δημιουργήθηκε με την βοήθεια του εγχειριδίου χρήσης του Pytorch και με την βοήθεια απλών μετρικών πάνω στα δεδομένα βλέπουμε στην Εικόνα 18 το ποσοστό επιτυχίας σε όλο το σύνολο δεδομένων επαλήθευσης (98%). Τα αποτελέσματα που φαίνονται στην Εικόνα 18 χρησιμοποιούν σαν μετρική τον απλό μέσο όρο (ΜΟ) στο σύνολο δεδομένων επαλήθευσης. Ο αλγόριθμος επαλήθευσης είναι ιδιαίτερα απλός, γίνεται επανάληψη στο σύνολο δεδομένων επαλήθευσης και δοκιμάζονται όλα τα δεδομένα πάνω στο μοντέλο. Όσα δεδομένα καταφέρνουν να αναγνωριστούν κατάλληλα από το μοντέλο συμβάλλουν θετικά στο ποσοστό επιτυχίας.



Εικόνα 18. Ποσοστό επιτυχίας μοντέλου στο σύνολο δεδομένων επαλήθευσης.

```
def get_total_network_performance(datasetmanager, model):
    model.eval();
    correct = 0
    total = len(datasetmanager.testloader.dataset)
    # since we're not training, we don't need to calculate the gradients for our outputs
    with torch.no_grad():
        for data in datasetmanager.testloader:
            images, labels = data
            # calculate outputs by running images through the network
            output = model(images)
            predicted = output.data.max(1, keepdim=True)[1]
            correct += predicted.eq(labels.data.view_as(predicted)).sum()
    print(f'Accuracy of the network on the 10000 test images: {100 * correct // total} %')
```

Εικόνα 19. Αλγόριθμος επαλήθευσης στο σύνολο δεδομένων επαλήθευσης.

Σε αυτό το σημείο έχει ολοκληρωθεί το κεφάλαιο που αναφέρθηκαν τα κατάλληλα εργαλεία με τα οποία δημιουργήθηκε η εφαρμογή, τις μεθόδους τις οποίες χρησιμοποιούμε για να εξάγουμε τα αποτελέσματα επεξήγησης του μοντέλου, την προεπεξεργασία που έγινε στα δεδομένα ελέγχου και εκπαίδευσης, ωστόσο περιγράφηκε το μοντέλο και κάθε διαδικασία που χρειάστηκε για να εκπαιδευτεί / επαληθευθεί. Επομένως έχουμε τα κατάλληλα εφόδια για να δούμε και να μελετήσουμε τα αποτελέσματα των μεθόδων εξήγησης του μοντέλου.

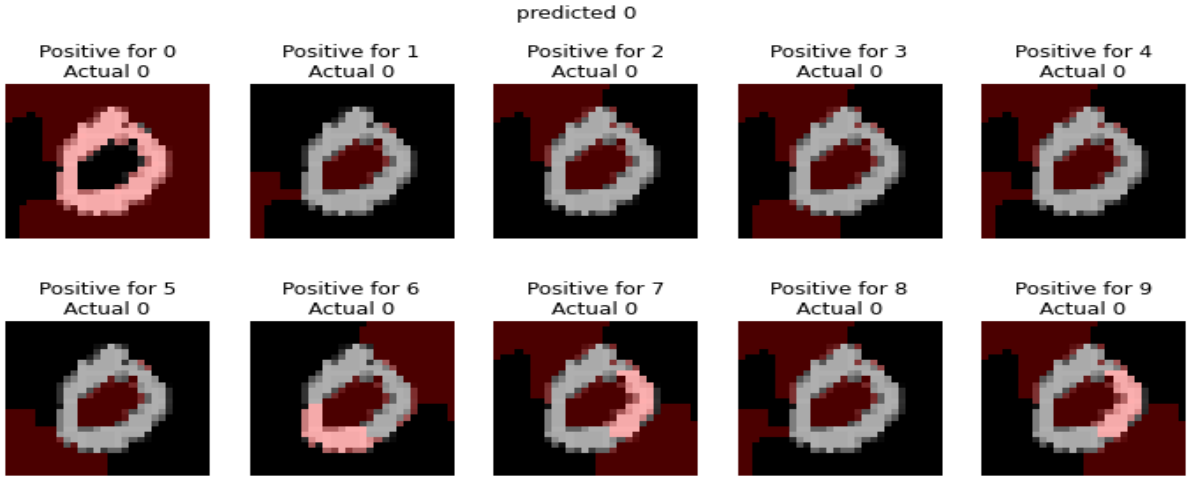
4 Αποτελέσματα

Σε αυτό το κεφάλαιο θα αναδείξουμε τα αποτελέσματα τα οποία λάβαμε από τις τεχνικές LIME και Deep SHAP με τις παραμέτρους που περιγράφονται στο προηγούμενο κεφάλαιο. Παράλληλα θα γίνεται και μια μικρή συζήτηση για να καταλάβουμε την μορφή των αποτελεσμάτων με την οποία τα παρατηρούμε. Κυρίως πρέπει να αναφέρουμε ότι θα αναδειχθούν δείγματα των αποτελεσμάτων και όχι με αναλυτική μορφή από όλο το σύνολο δεδομένων επαλήθευσης, τα αποτελέσματα τα οποία θα αναδειχθούν είναι δύο παραδείγματα για κάθε κλάση τα οποία επιλέχθηκαν μέσα από το σύνολο δεδομένων, αυτό θα γίνει για να υπάρχει μια καλή εικόνα των αποτελεσμάτων και των διαφόρων εξηγήσεων. Από ότι είδαμε στην Εικόνα 18. του κεφαλαίου 3 στην υποενότητα 3.5.2 δεν έχει κάθε κλάση την ίδια αποτελεσματικότητα και οι πιθανές εξήγησεις για κάθε κλάση, είναι πολύ πιθανό να αλλάζουν.

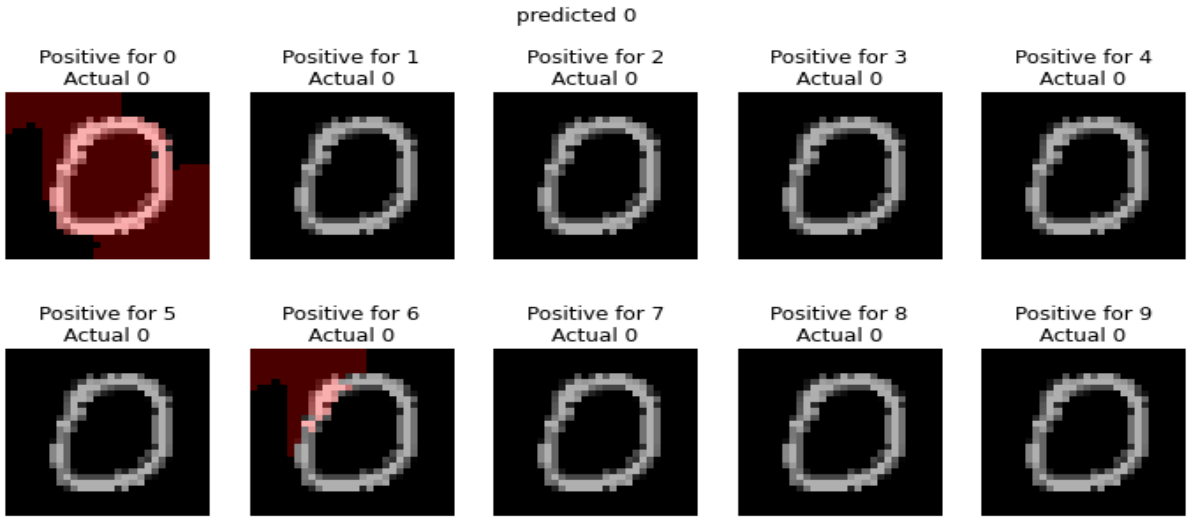
4.1 Αποτελέσματα LIME

Παρακάτω εμφανίζονται τα αποτελέσματα εξήγησης για κάθε κλάση τόσο για τις σωστές προβλέψεις που αναγνώρισε το νευρωνικό δίκτυο όσο και για τις λανθασμένες προβλέψεις. Στις παρακάτω εικόνες η εξήγηση φαίνεται μέσω των σκιασμένων, με κόκκινο χρώμα, περιοχών. Οι περιοχές αυτές αντιπροσωπεύουν τις θετικές συνεισφορές προς την πρόβλεψη του δικτύου. Απο τις σωστές προβλέψεις παρατηρούμε ότι το δίκτυο μας ανάλογα την περίπτωση του κάθε αριθμού εμφανίζει είτε αρκετές συνεισφορές για κάθε κλάση είτε πολύ λίγες (βλέπε Εικόνα 20). Αυτό μπορεί να συμβαίνει καθώς θα αναγνωρίζει ορισμένα σημαντικά στοιχεία που χρειάζεται μια κλάση για να ταιριάξει σε μια άλλη. Η περίπτωση της κλάσης μηδέν παρότι έχει αρκετές συνεισφορές παρατηρούμε ότι μόνο τα σημαντικά μέρη που αποτελούν τον αριθμό εμπεριέχονται στην εξήγηση για την ίδια την κλάση όποια άλλη κλάση μπορεί να μοιράζεται τυχόν όμοια σημεία (πχ απώλειας πληροφορίας) αλλά το σημαντικό κομμάτι που είναι ολόκληρος κύκλος δεν συνησφαίρει για τις υπόλοιπες κλάσεις. Ένα ακόμη σημαντικό χαρακτηριστικό που εμφανίζεται είναι ότι ανάλογα της περίπτωσης της εικόνας μπορεί τυχόν απώλεια πληροφορίας να μην συνεισφέρει θετικά στην μία περίπτωση αλλά σε κάποια άλλη εικόνα να συμπεριληφθεί η σημαντική απώλεια πληροφορίας (πχ περίπτωση εικόνων για κλάση μηδέν). Σε γενικές γραμμές τα αποτελέσματα των εικόνων φαίνονται αρκετά εμπιστεύσιμα και λογικά για την εξήγηση, όμως απο την περίπτωση της κλάσης πέντε παρατηρούμε ότι υπήρξε μια σύγχυση στην πρώτη εικόνα, ανάμεσα στις κλάσεις του πέντε, έξι και οκτώ όμως δεν ήταν μια ξεκάθαρη εικόνα για την κλάση πέντε ενώ στην πιο ευανάγνωστη εικόνα δεν υπάρχει αυτή η σύγχυση. Ενώ σε αντίθετη περίπτωση οι λανθασμένες εικόνες είναι κατα βάση εικόνες που λείπει πληροφορία στην εικόνα, πληροφορία που μπορεί να μην έλειπε κατα την εκπαίδευση, έτσι εμφανίζεται να είναι πιο κοντά σε άλλη κλάση (βλέπε αποτελέσματα κλάσης μηδέν). Επίσης αρκετές από τις εικόνες είναι ιδιαίτερα μη ευανάγνωστες ή ακόμα πολύ έντονες σε διάφορες περιοχές. Αυτό που φαίνεται ιδιαίτερα περίεργο σαν αποτέλεσμα είναι το αποτέλεσμα της κλάσης δύο, παρότι το δίκτυο πρόβλεψε την κλάση επτά η συνολική συνεισφορά στην κλάση του επτά έναντι της κλάσης δύο φαίνεται να είναι αρκετά μικρότερη. Για την τελευταία περίπτωση που αναφέρθηκε επειδή το αποτέλεσμα φαίνεται ιδιαίτερα περίεργο, θα δούμε για την ίδια εικόνα και την εξήγηση του Deep SHAP, μήπως εμφανίσει κάτι πιο σημαντικό για το δίκτυο.

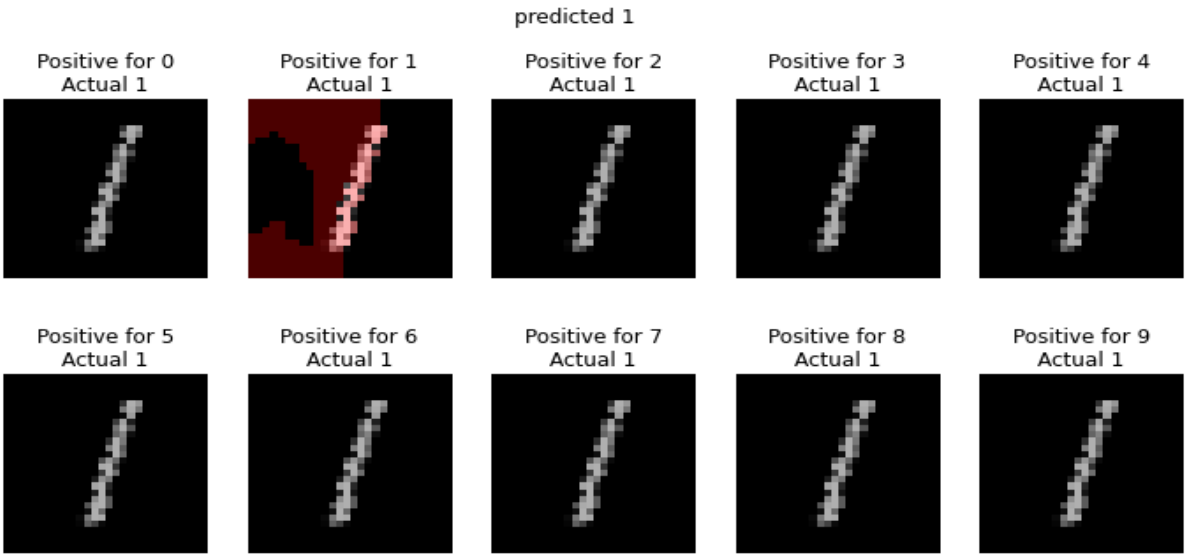
4.1.1 Σωστές προβλέψεις



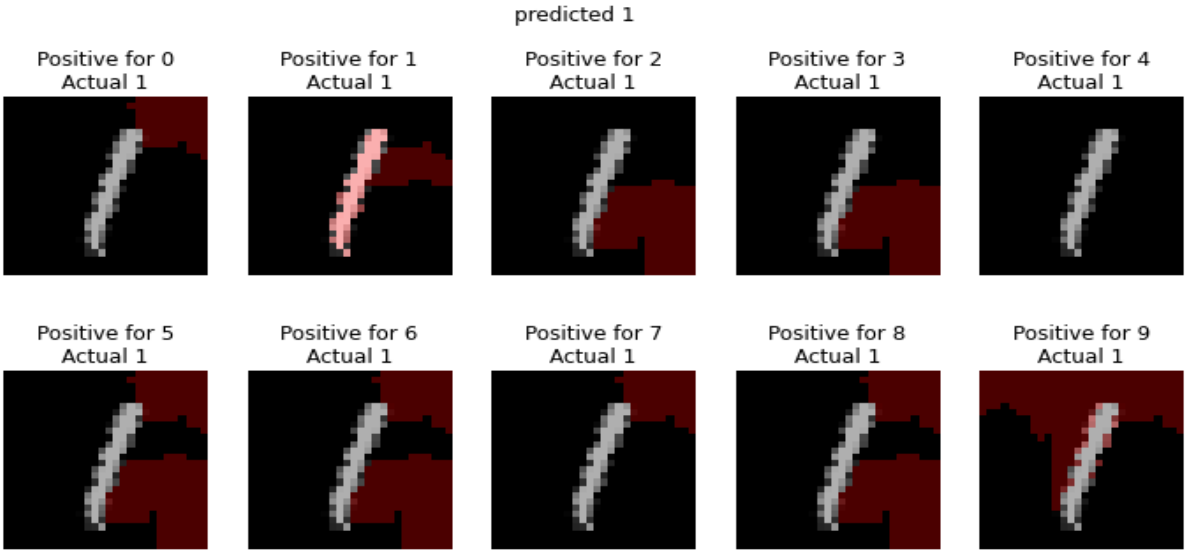
Εικόνα 20. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης μηδέν (0) με το LIME



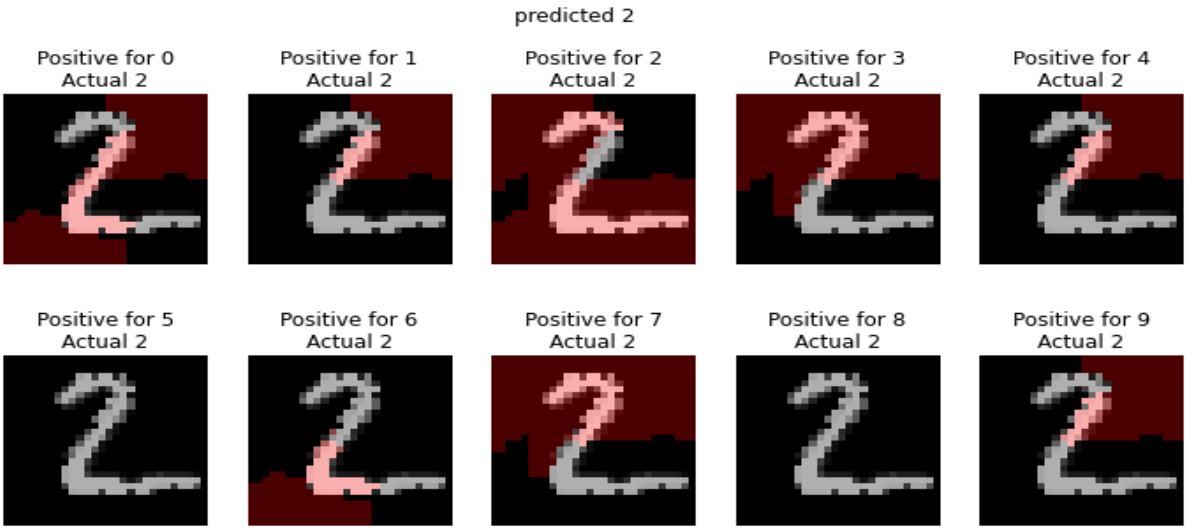
Εικόνα 21. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης μηδέν (0) με το LIME



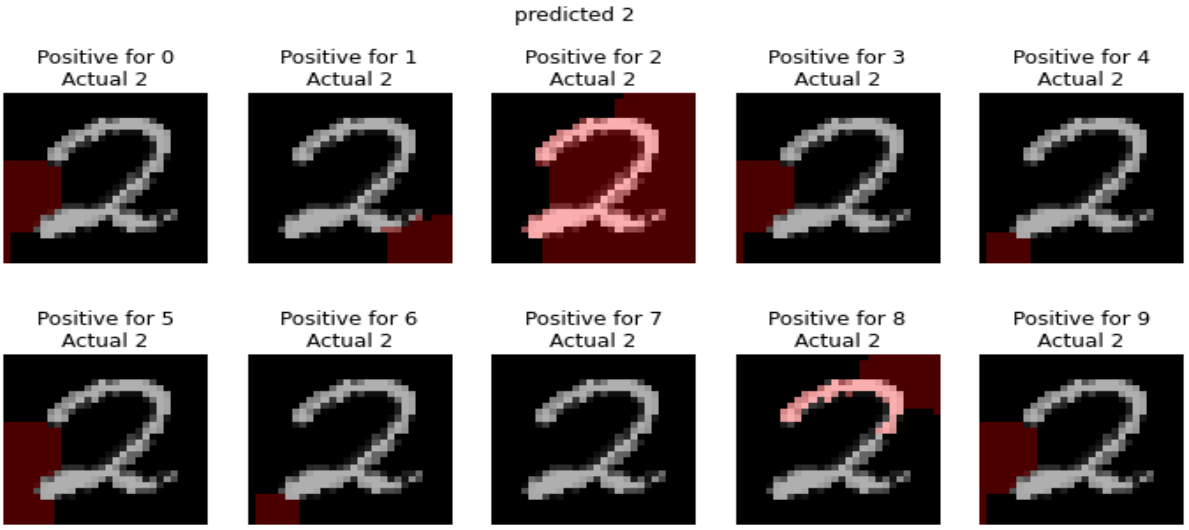
Εικόνα 22. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης ένα (1) με το LIME



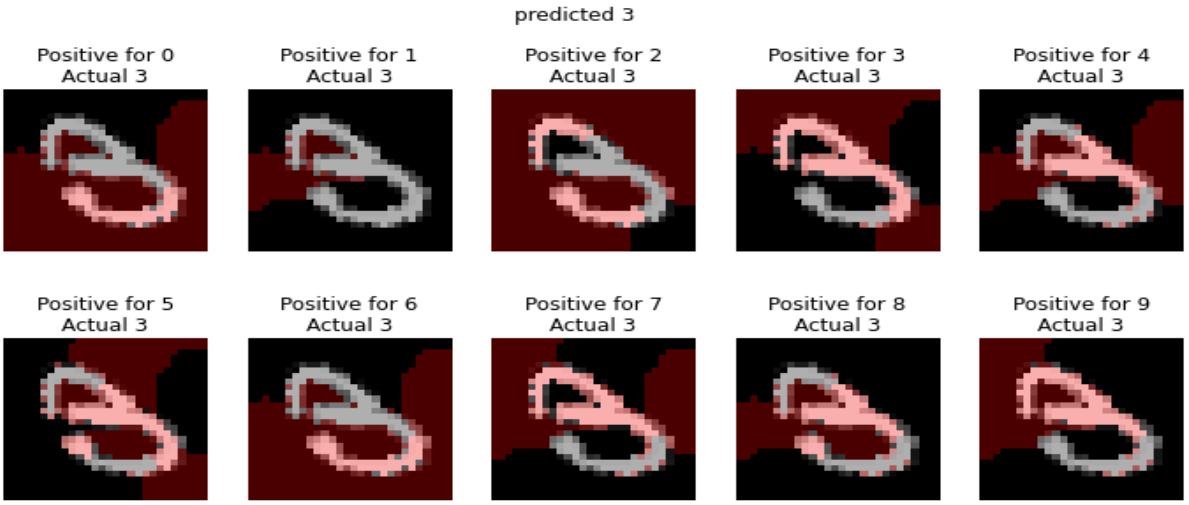
Εικόνα 23. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης ένα (1) με το LIME



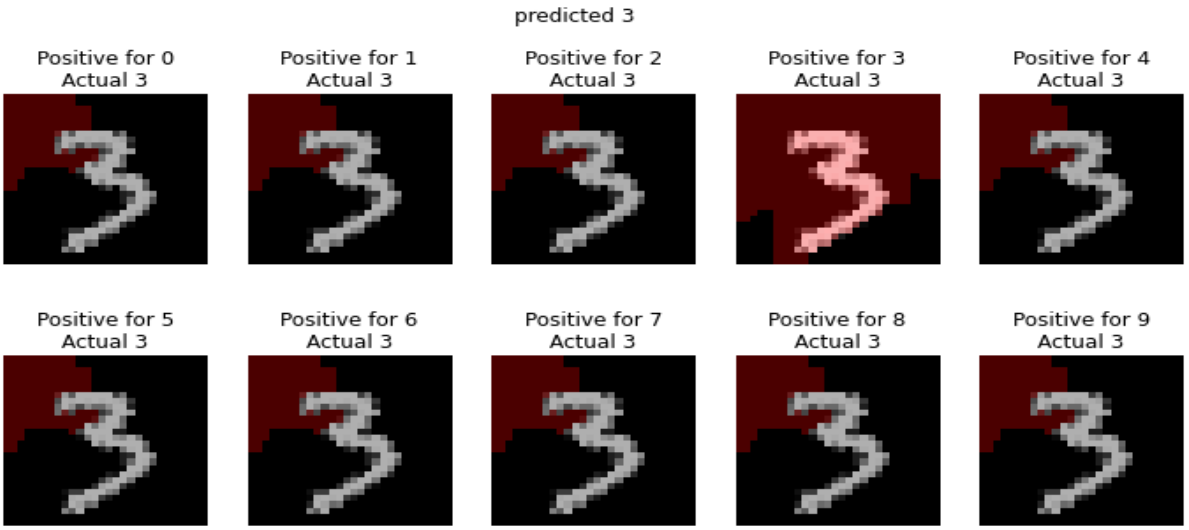
Εικόνα 24. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης δύο (2) με το LIME



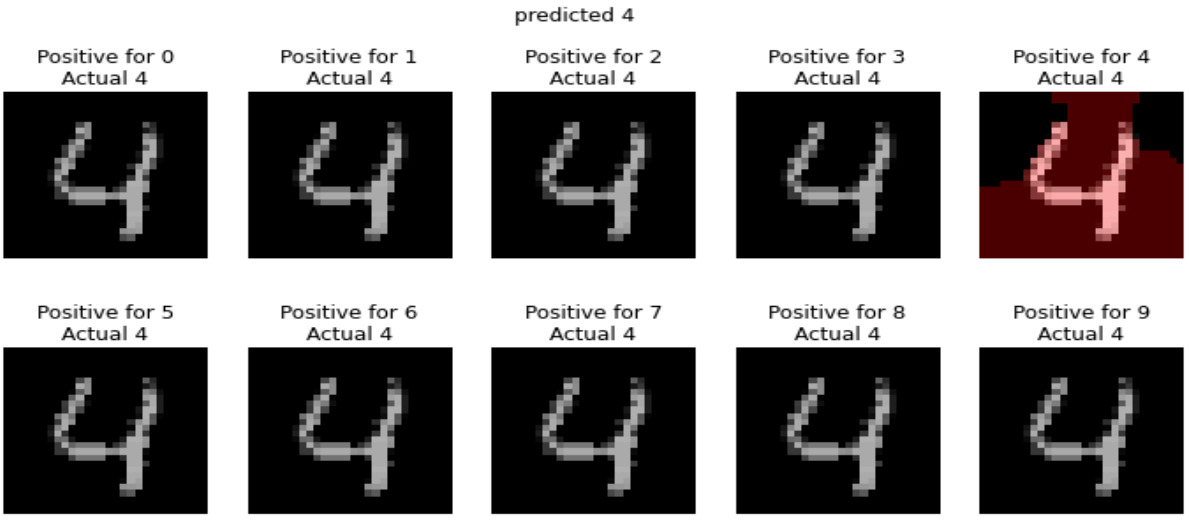
Εικόνα 25. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης δύο (2) με το LIME



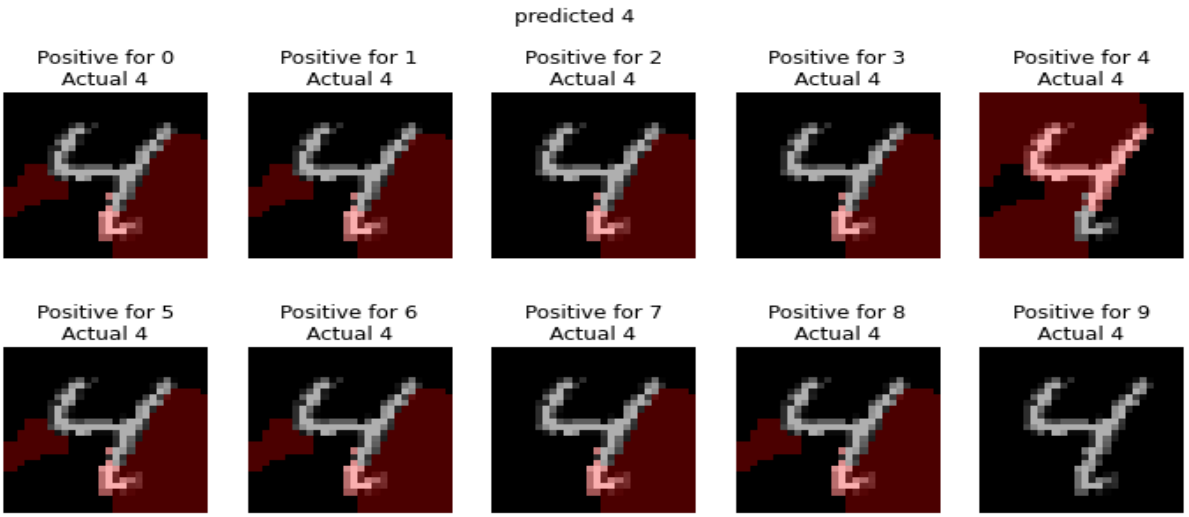
Εικόνα 26. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης τρία (3) με το LIME



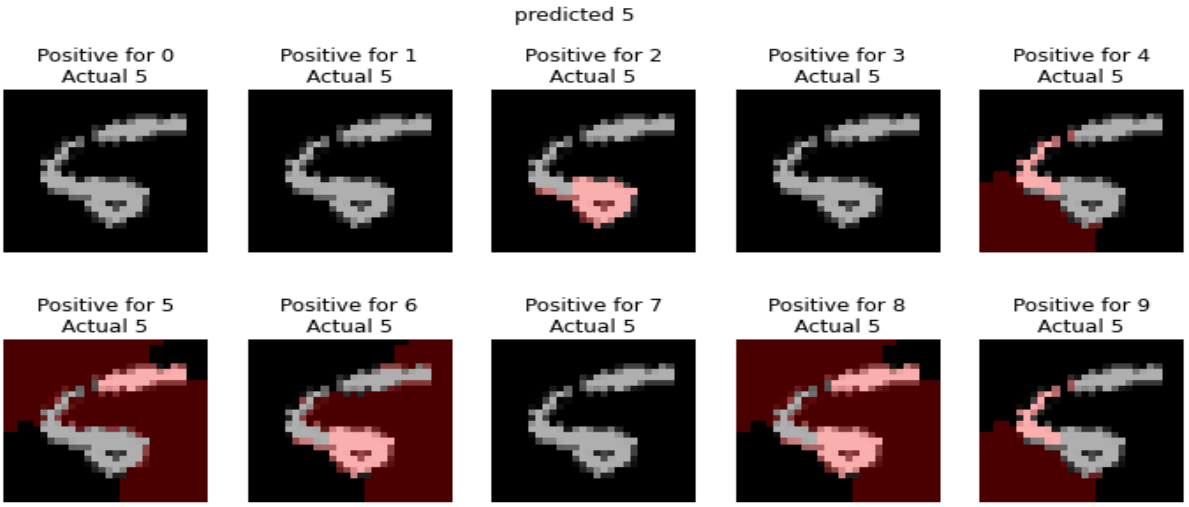
Εικόνα 27. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης τρία (3) με το LIME



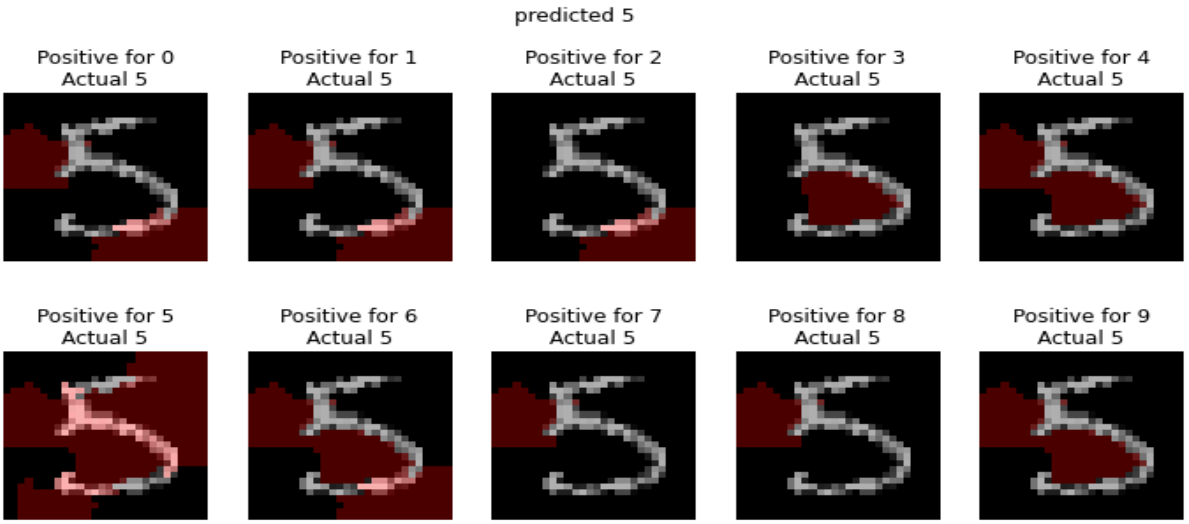
Εικόνα 28. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης τέσσερα (4) με το LIME



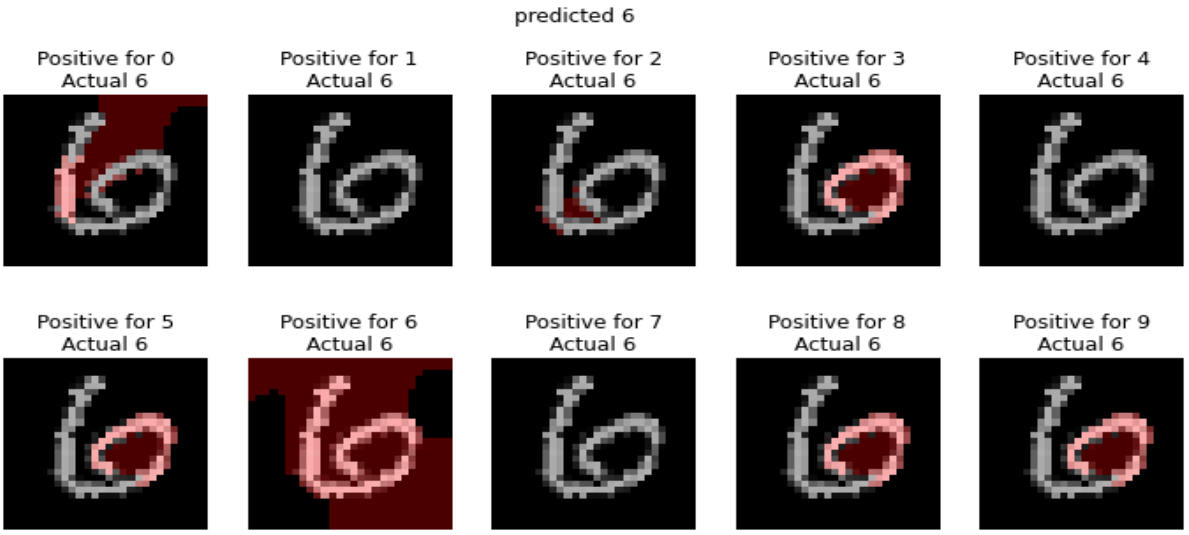
Εικόνα 29. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης τέσσερα (4) με το LIME



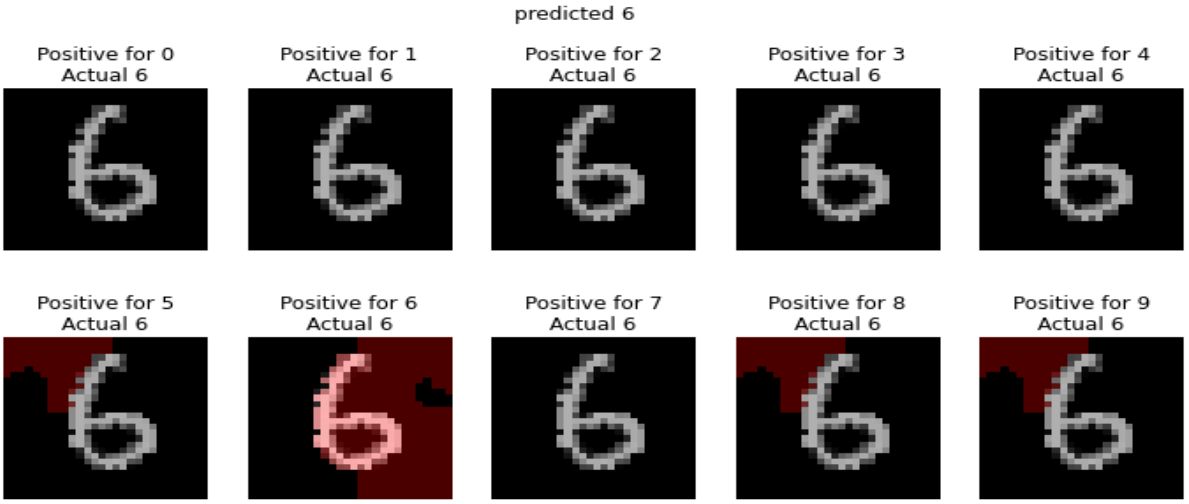
Εικόνα 30. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης πέντε (5) με το LIME



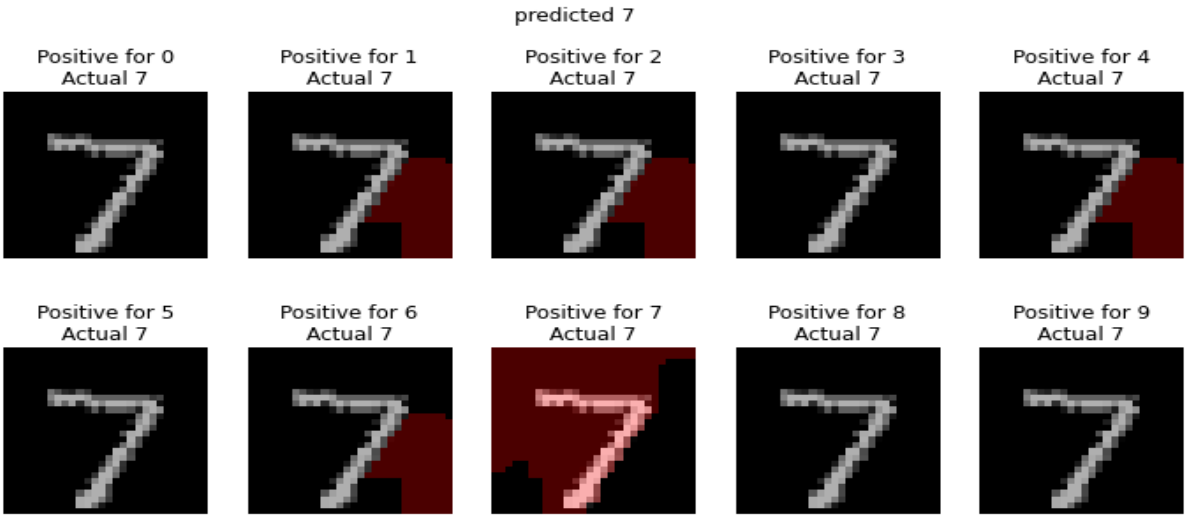
Εικόνα 31. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης πέντε (5) με το LIME



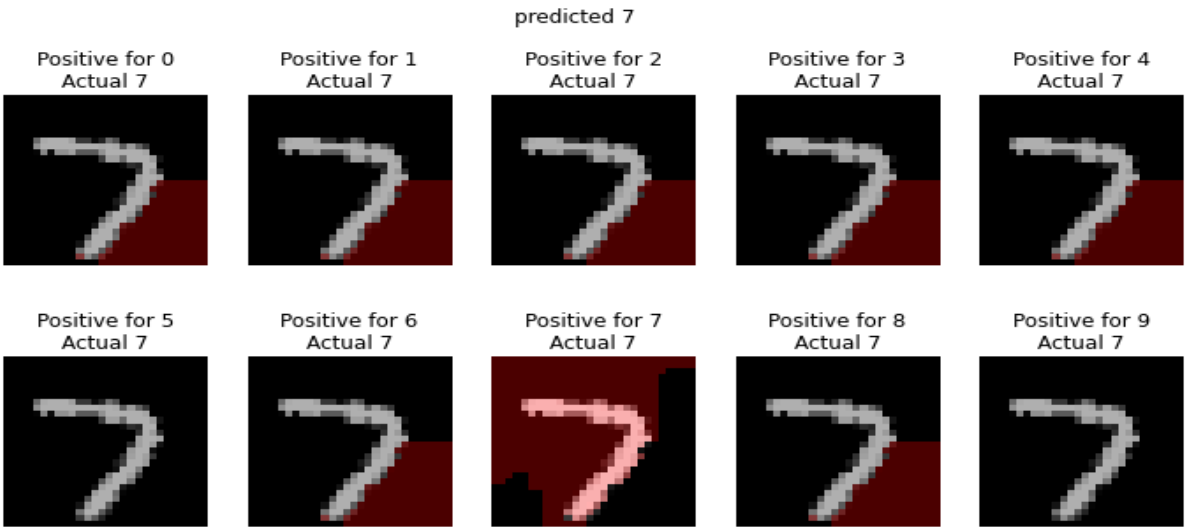
Εικόνα 32. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης έξη (6) με το LIME



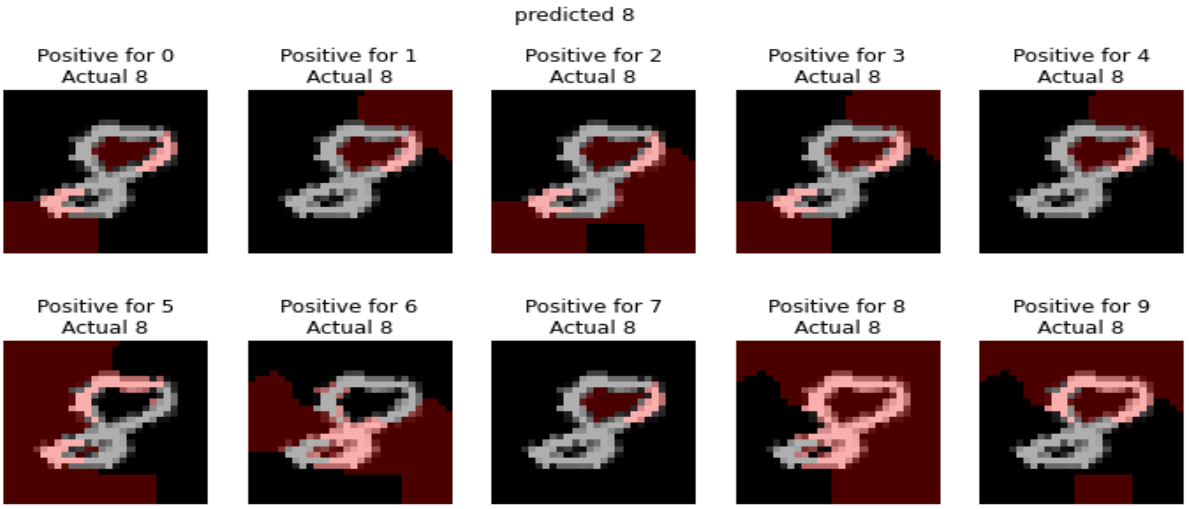
Εικόνα 33. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης έξη (6) με το LIME



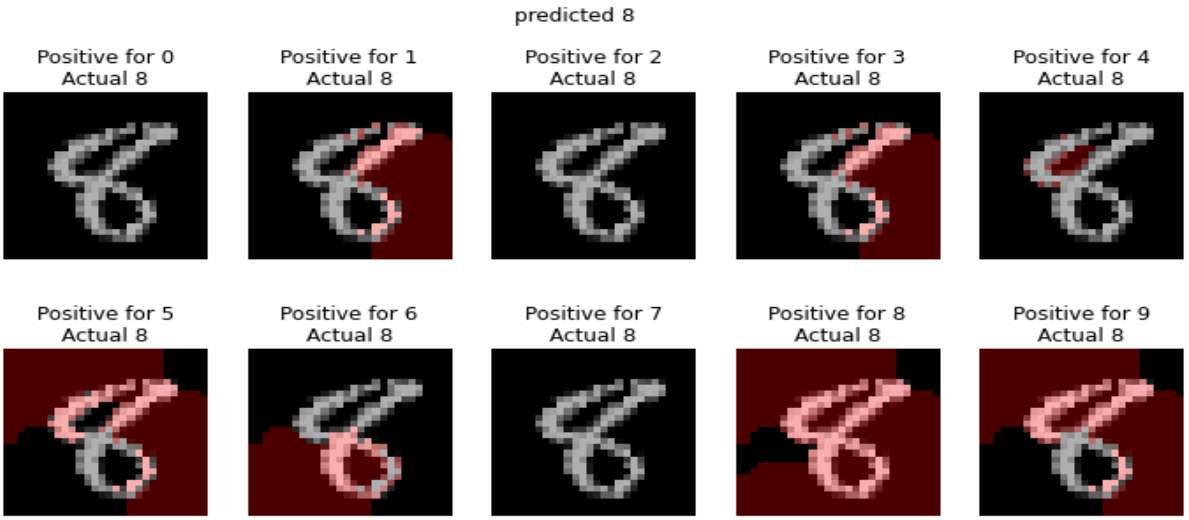
Εικόνα 34. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης επτά (7) με το LIME



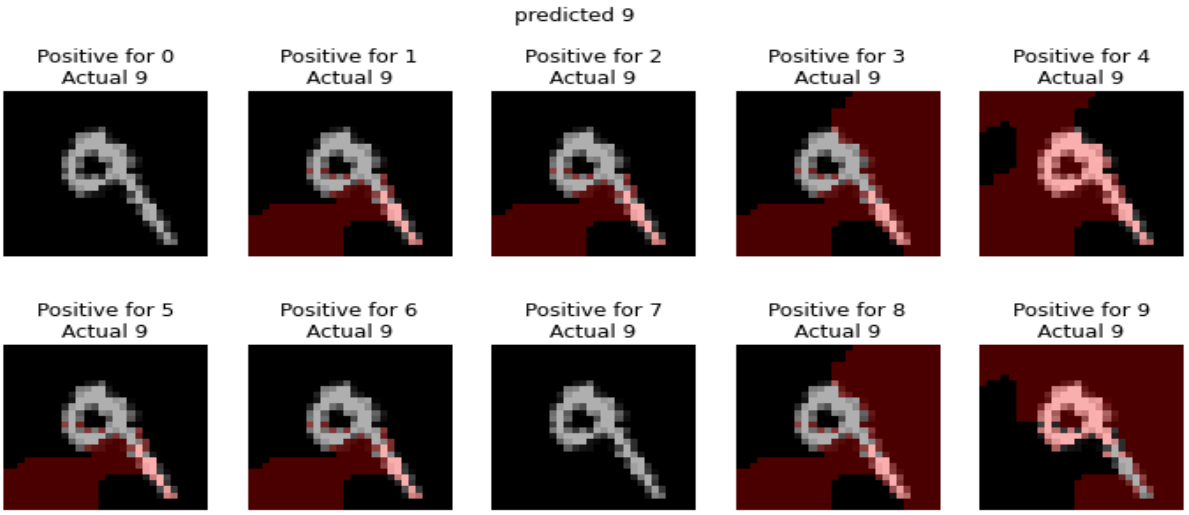
Εικόνα 35. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης επτά (7) με το LIME



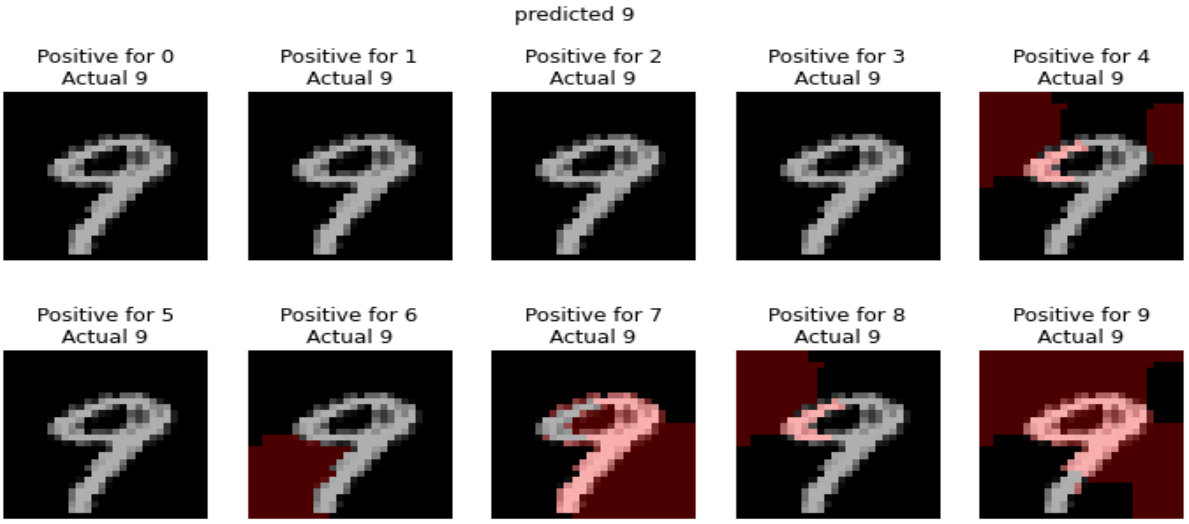
Εικόνα 36. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης οκτώ (8) με το LIME



Εικόνα 37. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης οκτώ (8) με το LIME

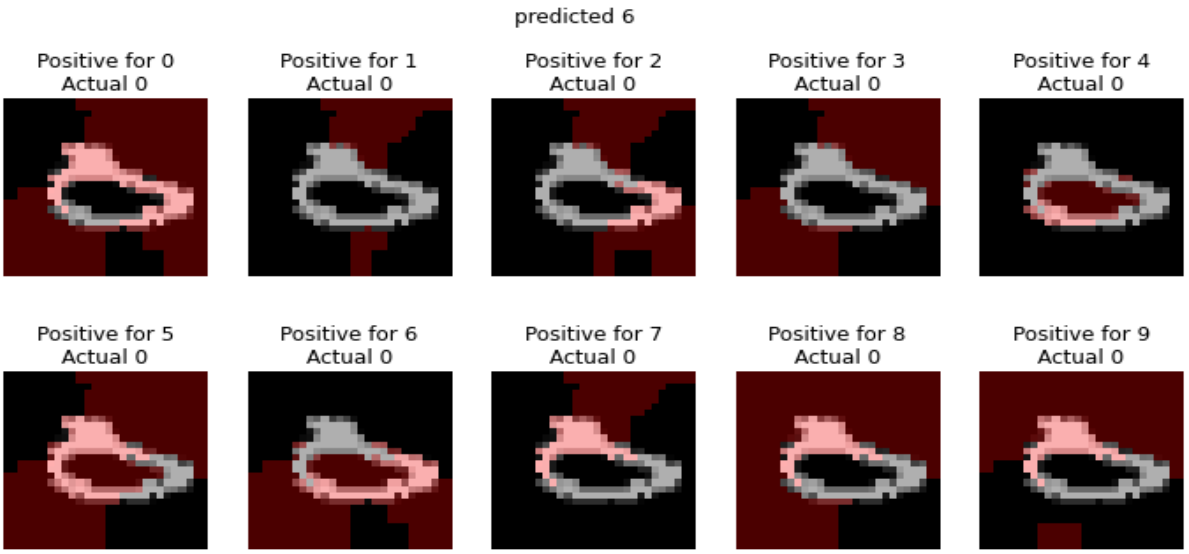


Εικόνα 38. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης εννέα (9) με το LIME

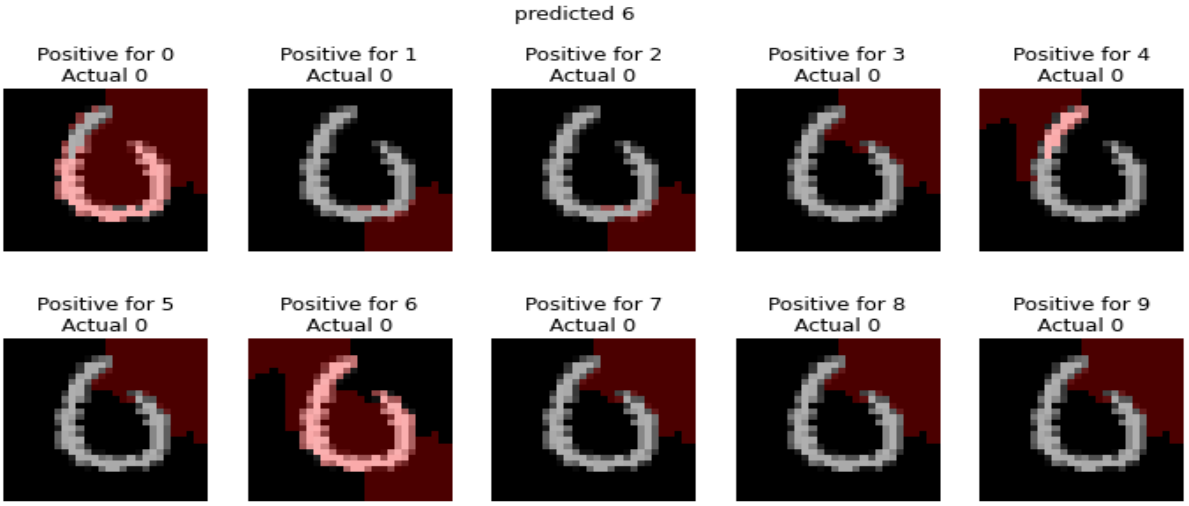


Εικόνα 39. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης εννέα (9) με το LIME

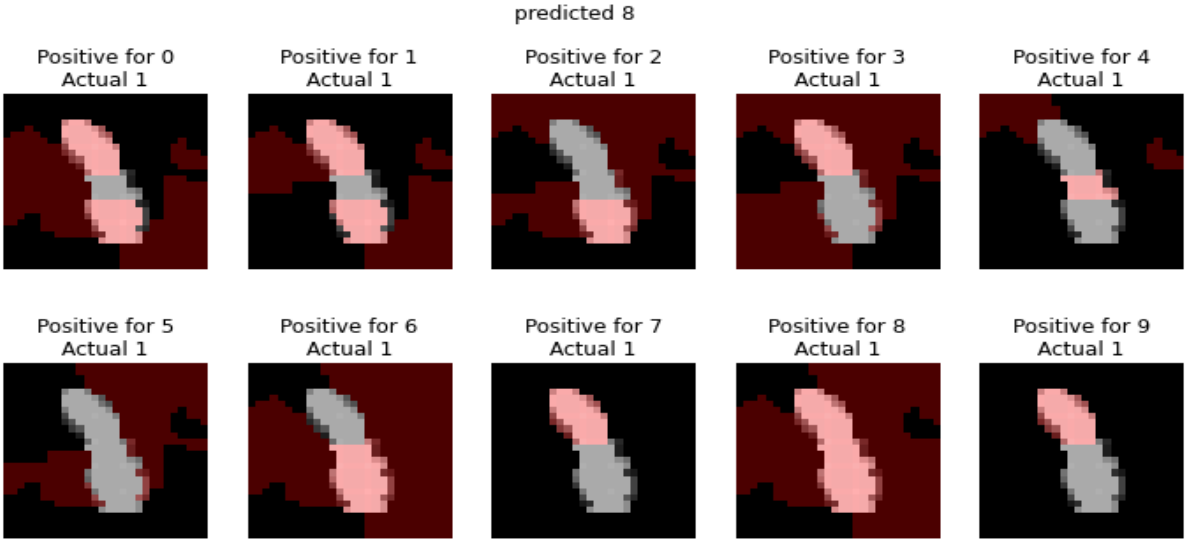
4.1.2 Λανθασμένες προβλέψεις



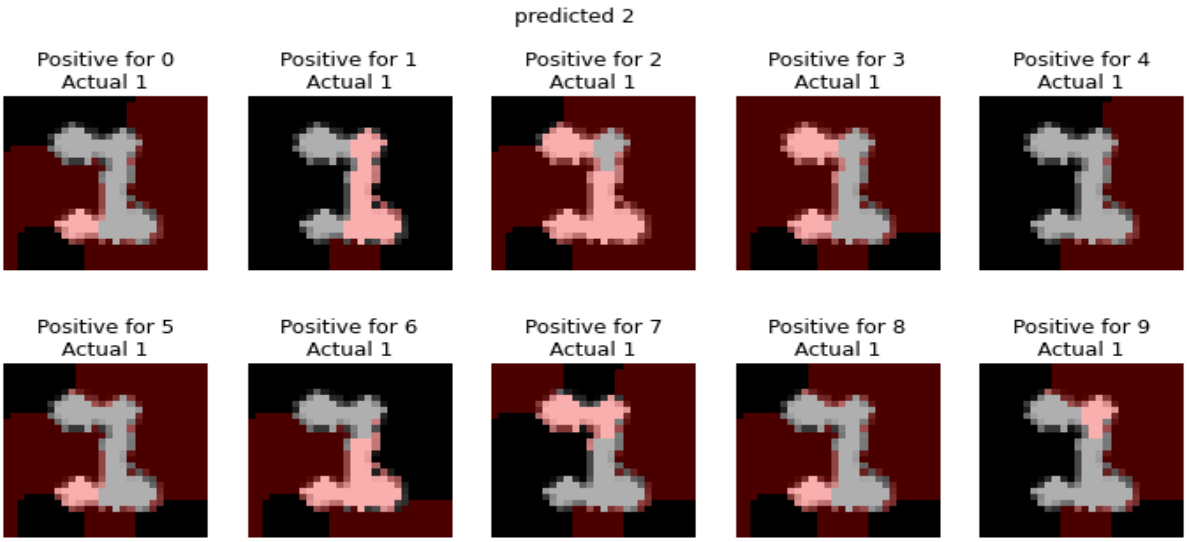
Εικόνα 40. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης μηδέν (0) με το LIME



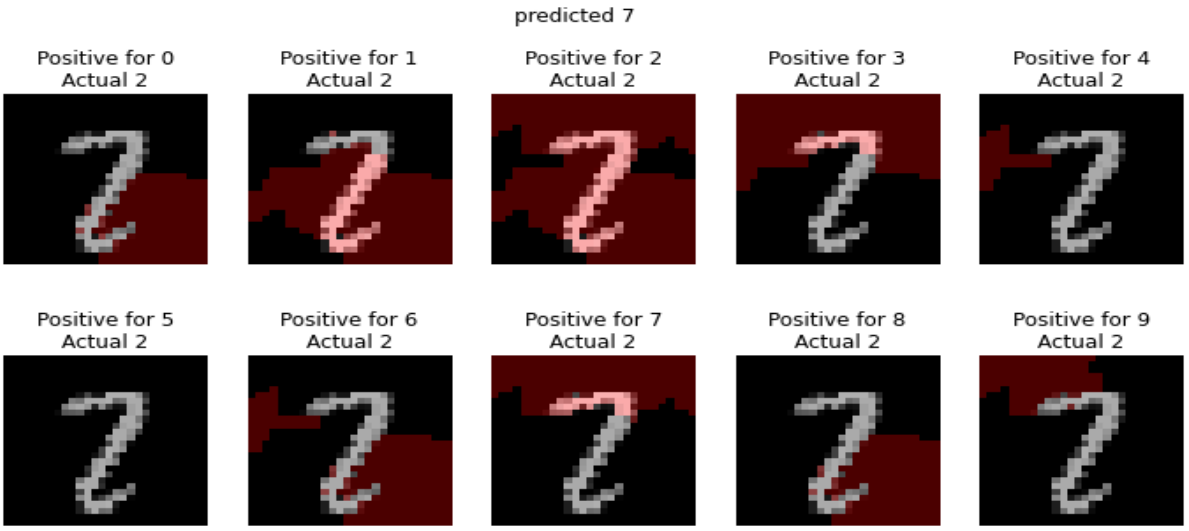
Εικόνα 41. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης μηδέν (0) με το LIME



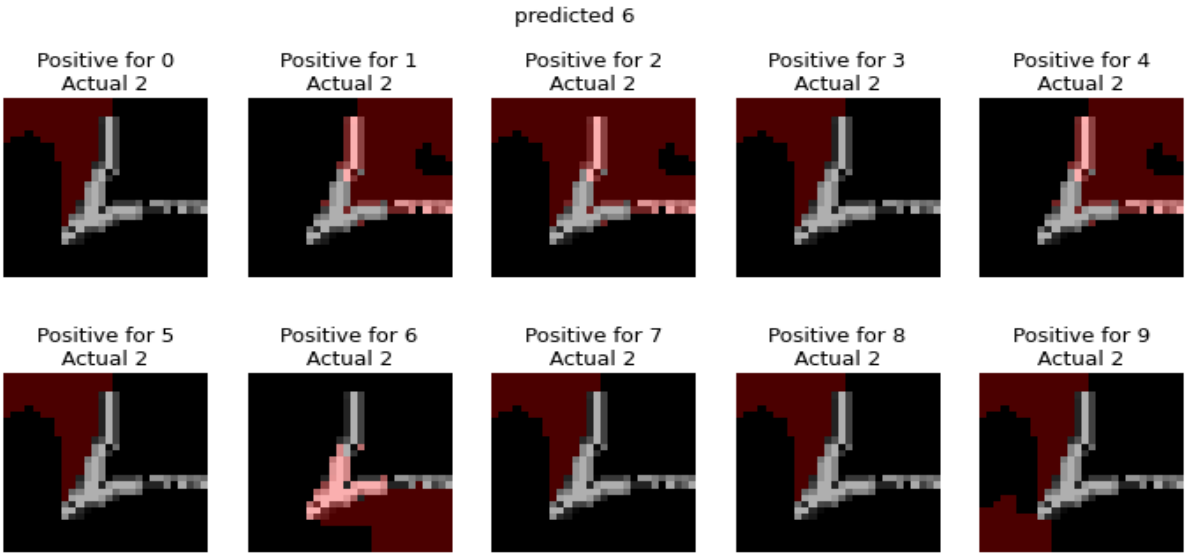
Εικόνα 42. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης ένα (1) με το LIME



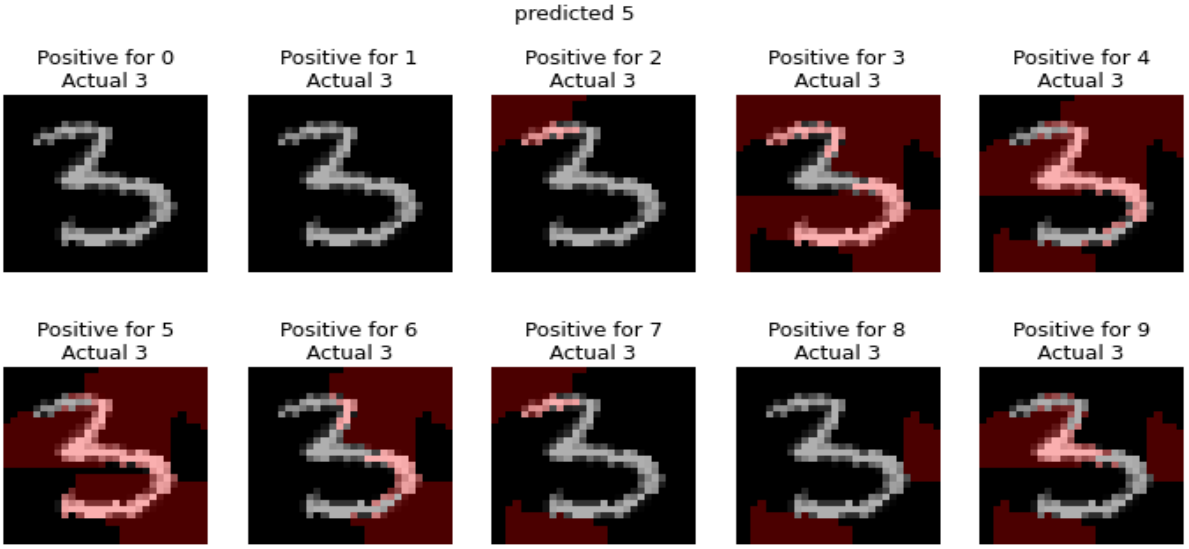
Εικόνα 43. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης ένα (1) με το LIME



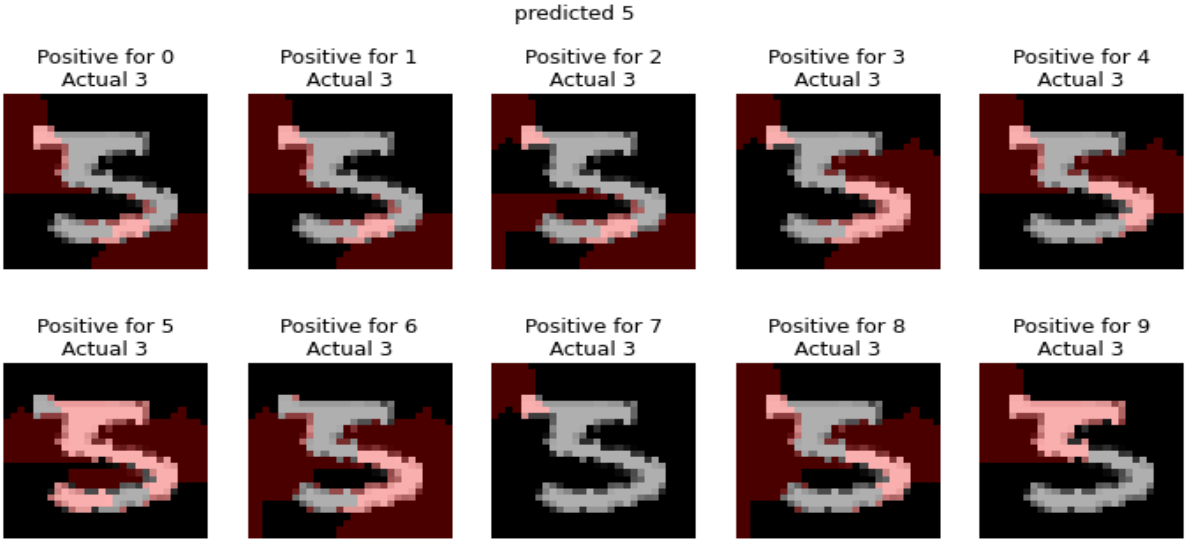
Εικόνα 44. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης δύο (2) με το LIME



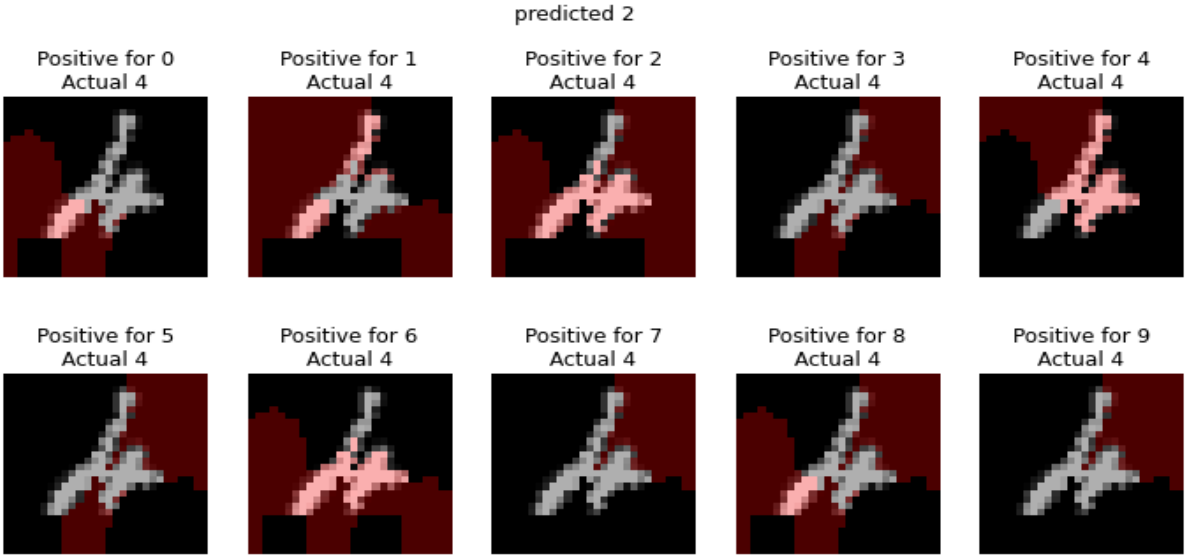
Εικόνα 45. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης δύο (2) με το LIME



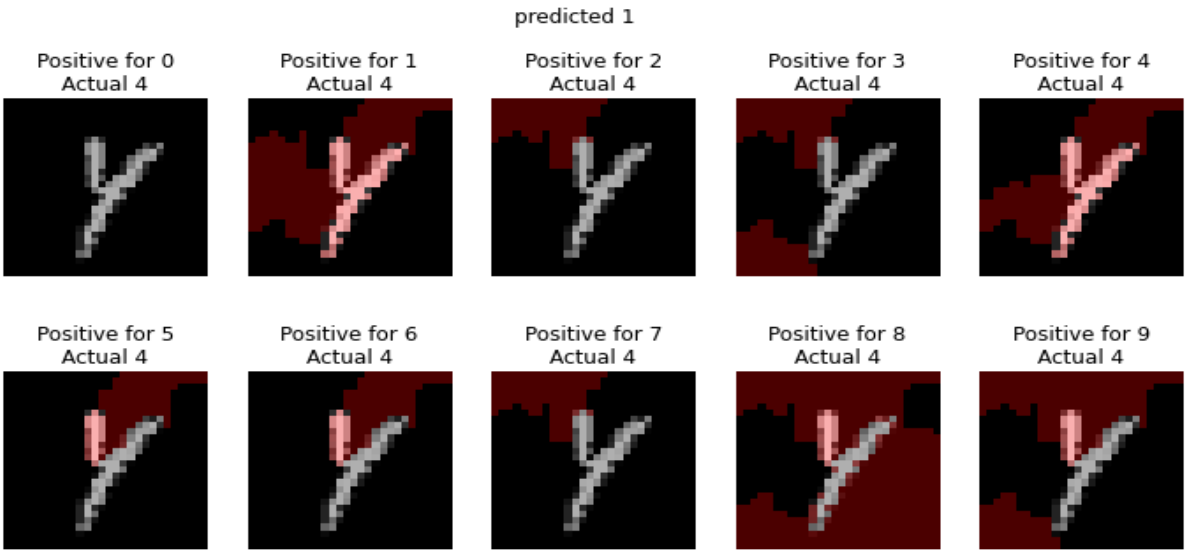
Εικόνα 46. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης τρία (3) με το LIME



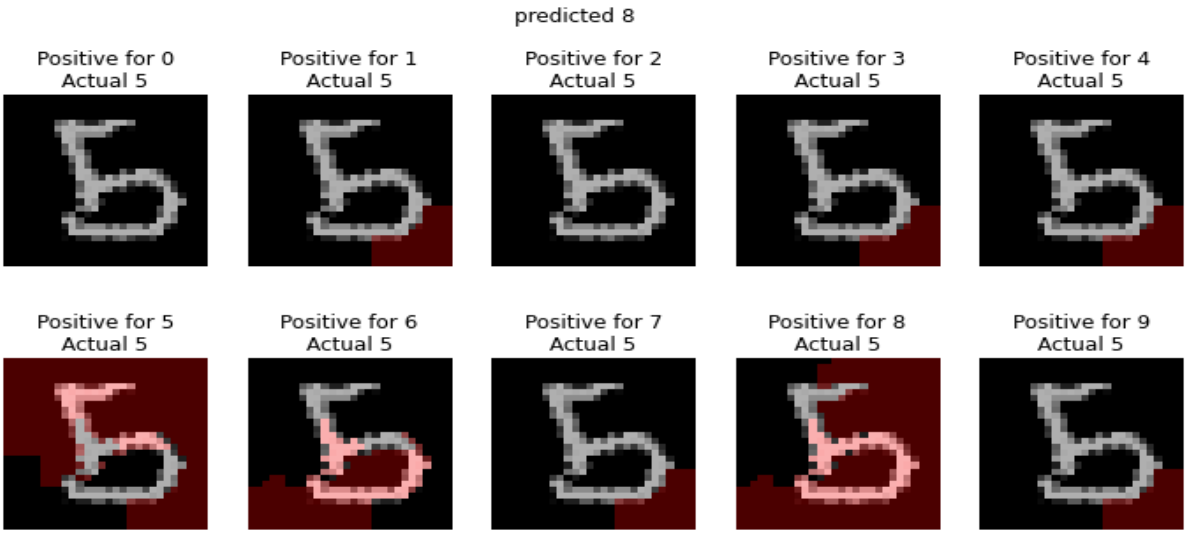
Εικόνα 47. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης τρία (3) με το LIME



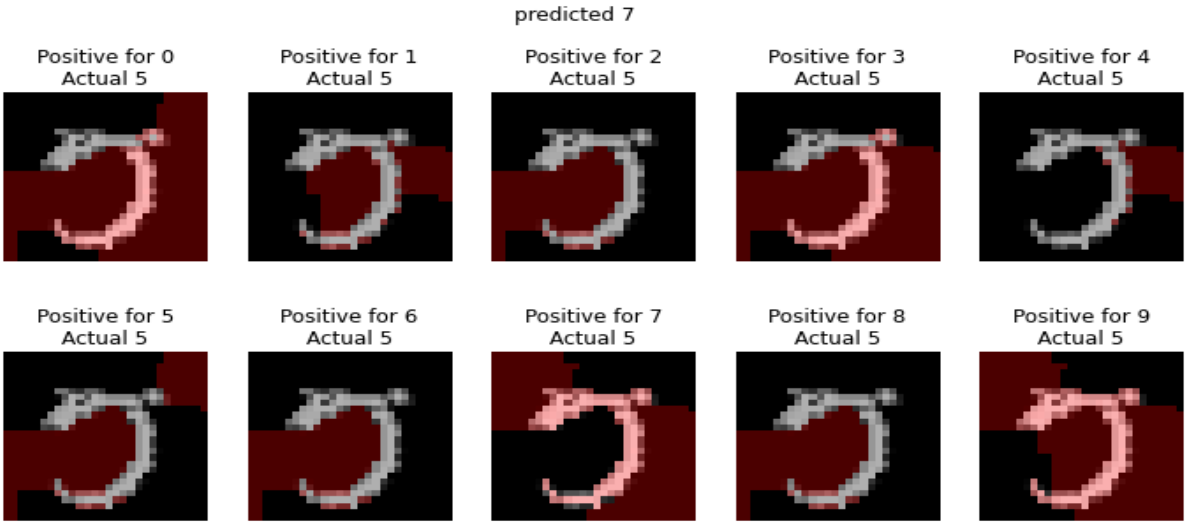
Εικόνα 48. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης τέσσερα (4) με το LIME



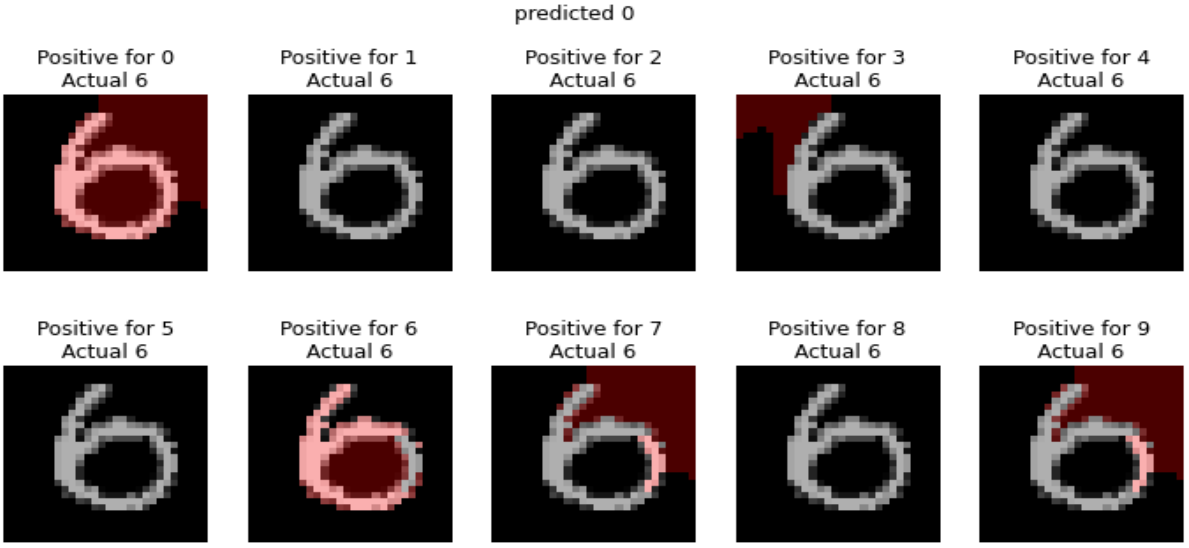
Εικόνα 49. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης τέσσερα (4) με το LIME



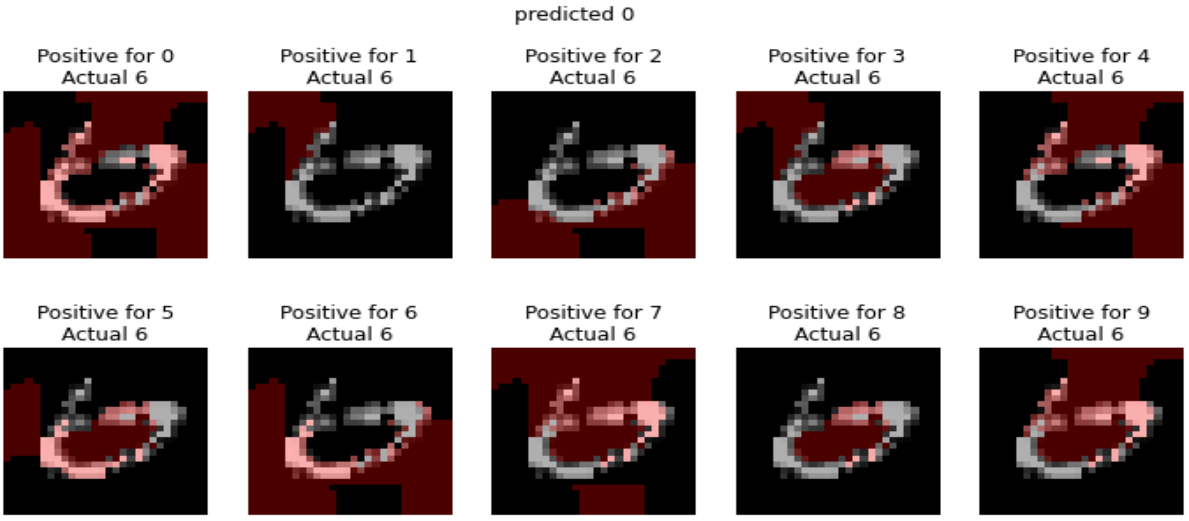
Εικόνα 50. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης πέντε (5) με το LIME



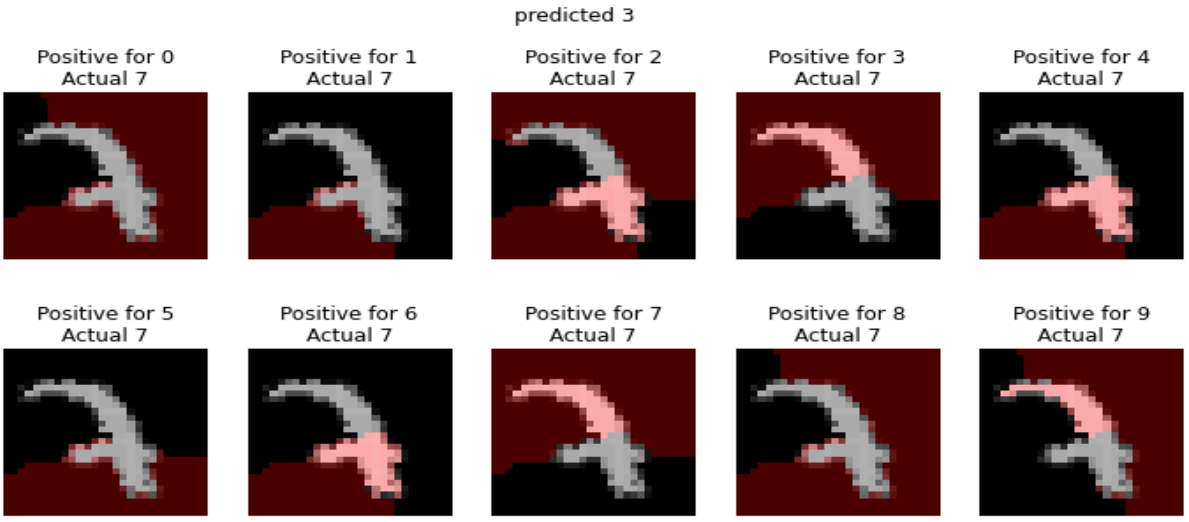
Εικόνα 51. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης πέντε (5) με το LIME



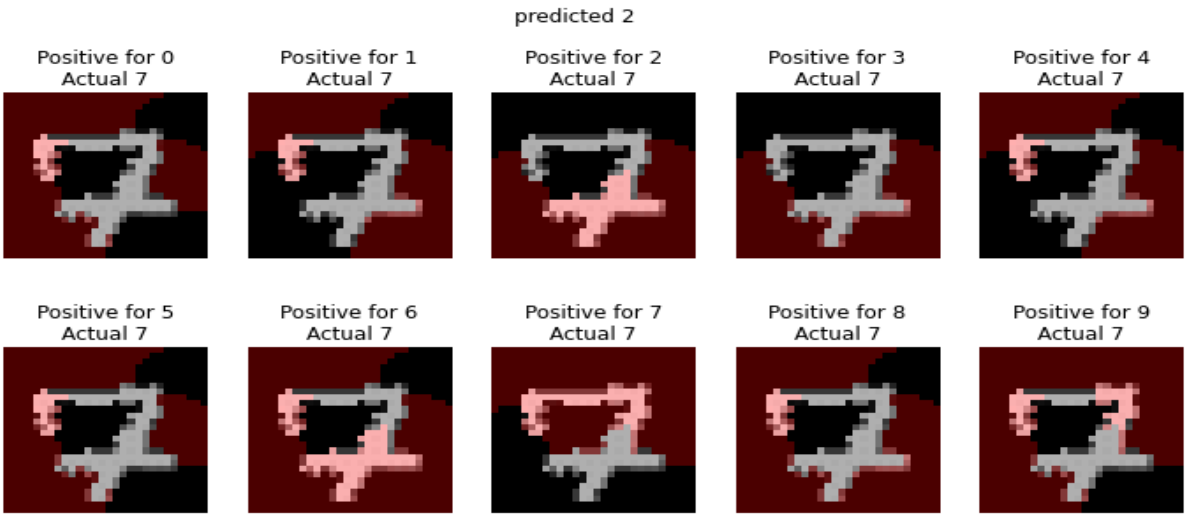
Εικόνα 52. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης έξη (6) με το LIME



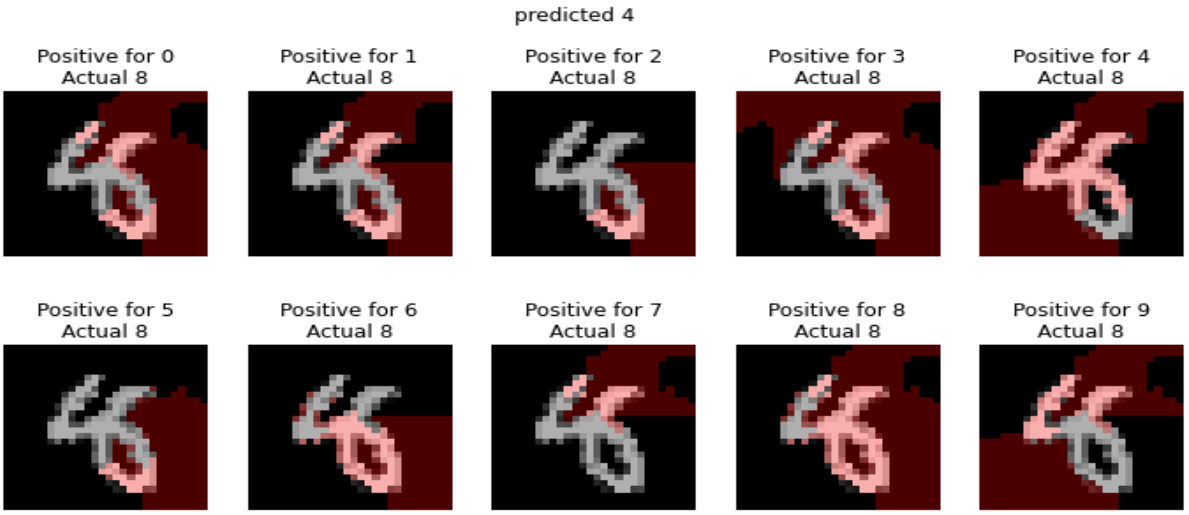
Εικόνα 53. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης έξη (6) με το LIME



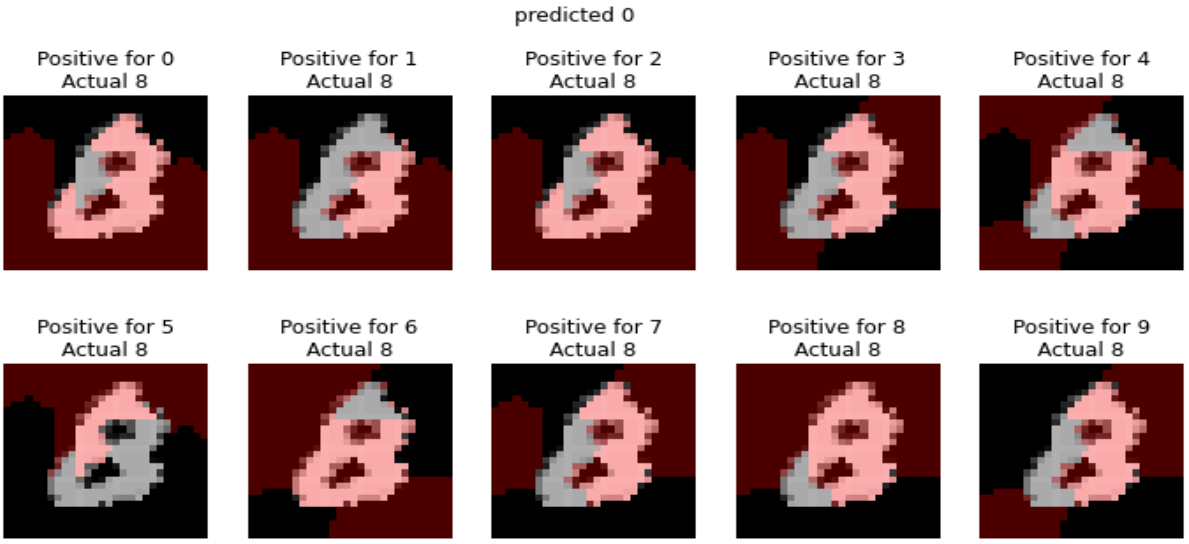
Εικόνα 54. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης επτά (7) με το LIME



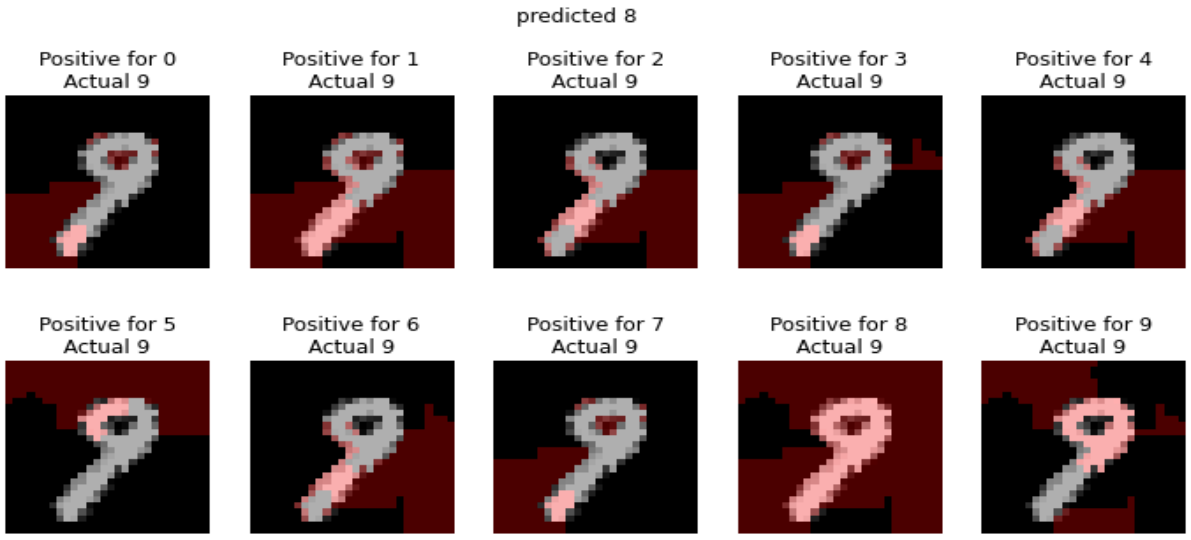
Εικόνα 55. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης επτά (7) με το LIME



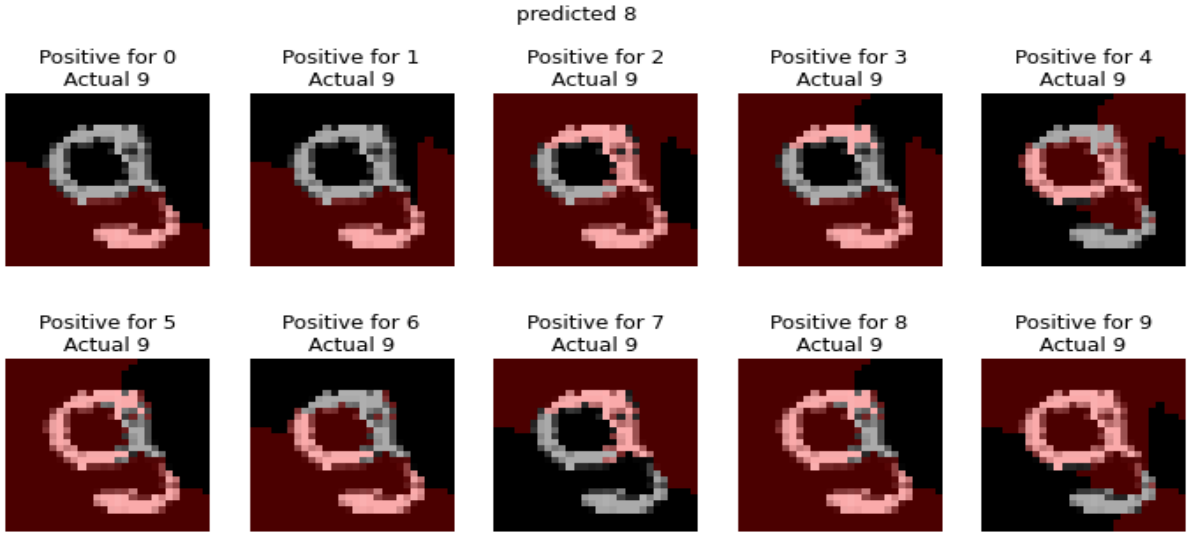
Εικόνα 56. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης οκτώ (8) με το LIME



Εικόνα 57. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης οκτώ (8) με το LIME



Εικόνα 57. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης εννέα (9) με το LIME



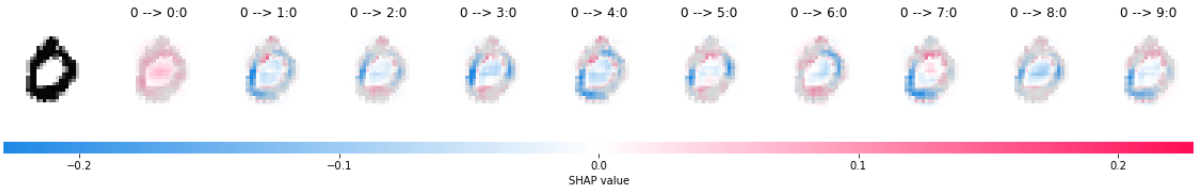
Εικόνα 58. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης εννέα (9) με το LIME

4.2 Αποτελέσματα Deep SHAP

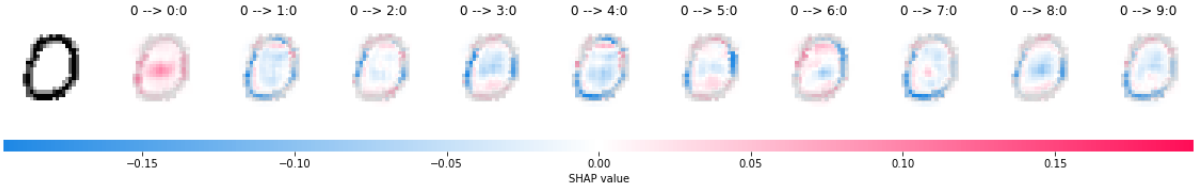
Παρακάτω εμφανίζονται τα αποτελέσματα εξήγησης για κάθε κλάση τόσο για τις σωστές προβλέψεις που αναγνώρισε το νευρωνικό δίκτυο όσο και για τις λανθασμένες προβλέψεις. Στα αποτελέσματα εμφανίζονται οι θετικές συνεισφορές με κόκκινο χρώμα ενώ οι αρνητικές συνεισφορές εμφανίζονται με μπλε χρώμα. Οι εικόνες εμφανίζουν από κάτω τους και την κλίμακα για την φωτεινότητα των χρωμάτων. Απο τις σωστές προβλέψεις παρατηρούμε ότι το νευρωνικό δίκτυο φαίνεται να έχει μάθει αρκετά για τους αριθμούς και το πως μοιάζουν, αυτο που φαίνεται ενδιαφέρον είναι ότι και το Deep SHAP έχει βγάλει ότι το δίκτυο συμπεριλαμβάνει θετικά διάφορες περιοχές που δεν υπάρχει πληροφορία. Έτσι στα διάφορα αποτελέσματα για κάθε κλάση εάν ένας αριθμός δεν συμπεριλαμβάνει και δεν θα έπρεπε να συμπεριλάβει σε μια περιοχή πληροφορία φαίνεται ότι για τις άλλες κλάσεις ίσως αφαιρεθεί, βλέπε αποτελέσματα της κλάσης τέσσερα όπου το πάνω κομμάτι του αριθμού συμβάλει αρνητικά για την κλάση εννέα λόγω απώλειας πληροφορίας. Ενώ απο τις λανθασμένες προβλέψεις παρατηρούμε ότι οι περιπτώσεις δεν είναι τόσο ευανάγνωστες και αρκετά

συχνά προκαλούν ένα τύπου σύγχυσης. Σε αυτή την σύγχυση όμως φαίνεται να υπάρχουν σοβαρές ενδείξεις για την πρόβλεψη της σωστής κλάσης όμως υπήρχαν χαρακτηριστικά για τα οποία συνήσφεραν θετικά σε μία άλλη κλάση και για αυτό έβγαινε η λανθασμένη κλάση, ένα τέτοιο παράδειγμα είναι και η κλάση δύο που μας φάνηκε περίεργο στην προηγούμενη μέθοδο. Ωστόσο υπάρχουν και περιπτώσεις όπου απλά υπήρχαν χαρακτηριστικά που έμοιαζαν περισσότερο σε άλλη κλάση από αυτό που ήταν πραγματικά (βλέπε παράδειγμα λανθασμένων προβλέψεων κλασης τρία (3)). Γενικα οι εξηγήσεις φαίνονται να είναι αρκετά αληθοφανείς για το τι κοιτάει το δίκτυο πραγματικα και φαίνεται να έχει μάθει διάφορα χαρακτηριστικά.Τα αποτελέσματα μας έδειξαν οτι μορφές των εικόνων παίζουν σημαντικό ρόλο, καθώς μπορεί να μοιάζουν περισσότερο σε μια άλλη κλάση παρά στην πραγματική. Αυτό μπορεί να συμβαίνει λόγω της μορφής των δεδομένων από τα οποία εκπαιδεύτηκε, πρακτικά μπορεί να είναι πολύ πιο ευανάγνωστα, ίσως με καλύτερη εκπαίδευση του δικτύου αυτές οι καταστάσεις να ξεπεραστούν.

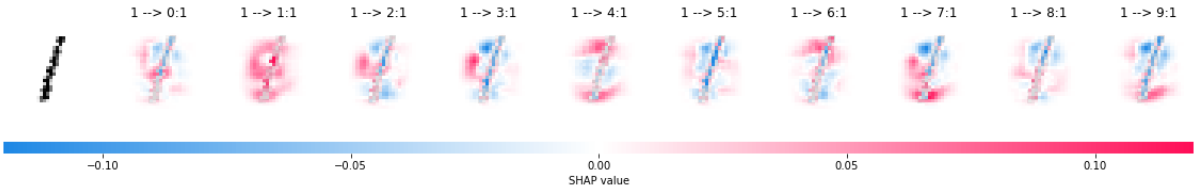
4.2.1 Σωστές προβλέψεις



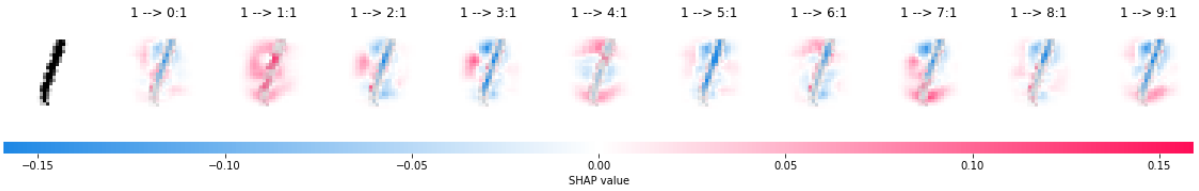
Εικόνα 59. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης μηδέν (0) με το Deep SHAP



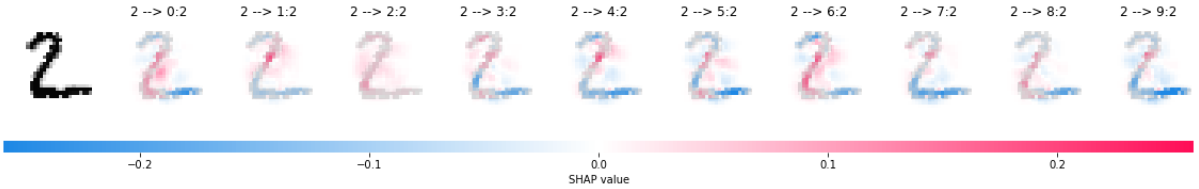
Εικόνα 60. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης μηδέν (0) με το Deep SHAP



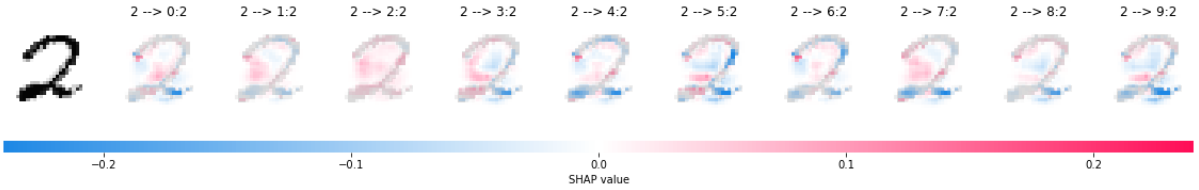
Εικόνα 61. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης ένα (1) με το Deep SHAP



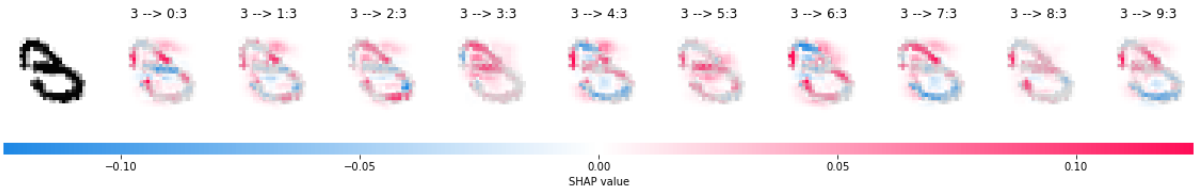
Εικόνα 62. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης ένα (1) με το Deep SHAP



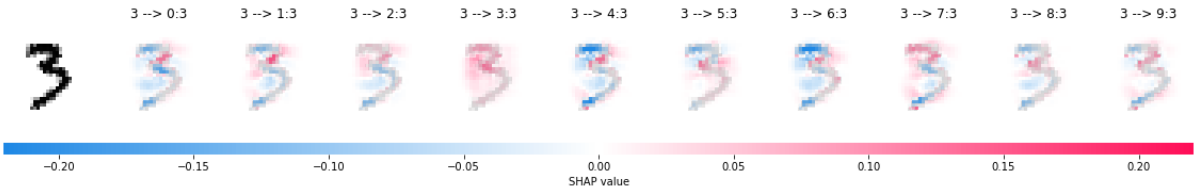
Εικόνα 63. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης δύο (2) με το Deep SHAP



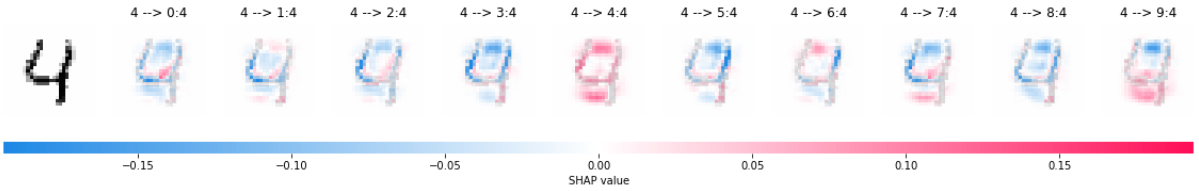
Εικόνα 64. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης δύο (2) με το Deep SHAP



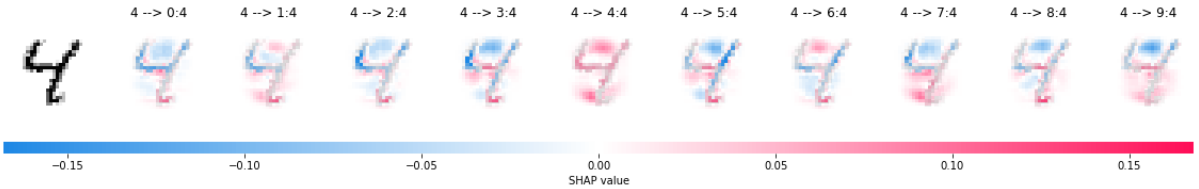
Εικόνα 65. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης τρία (3) με το Deep SHAP



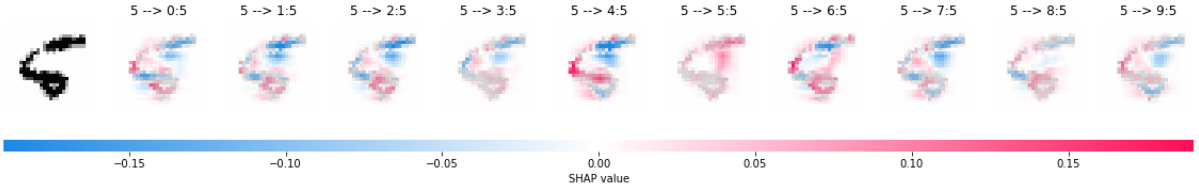
Εικόνα 66. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης τρία (3) με το Deep SHAP



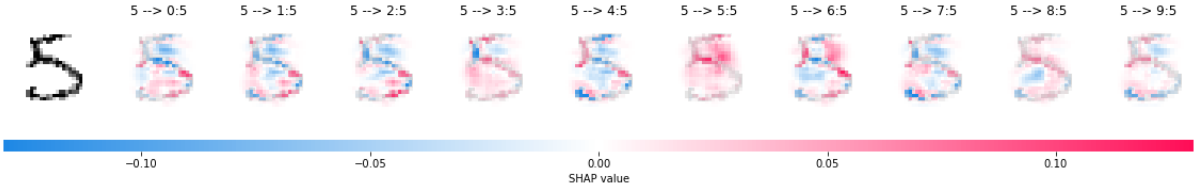
Εικόνα 67. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης τέσσερα (4) με το Deep SHAP



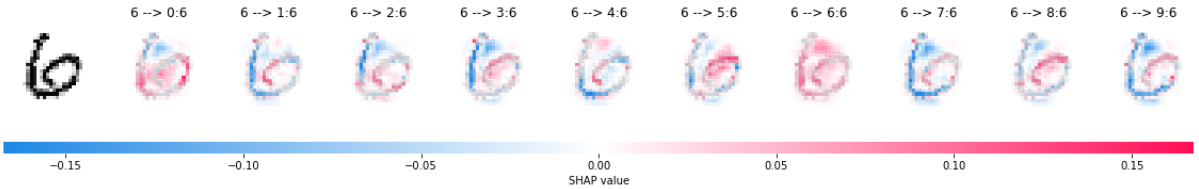
Εικόνα 68. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης τέσσερα (4) με το Deep SHAP



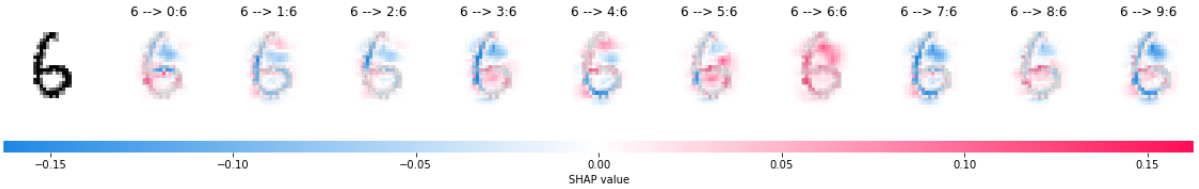
Εικόνα 69. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης πέντε (5) με το Deep SHAP



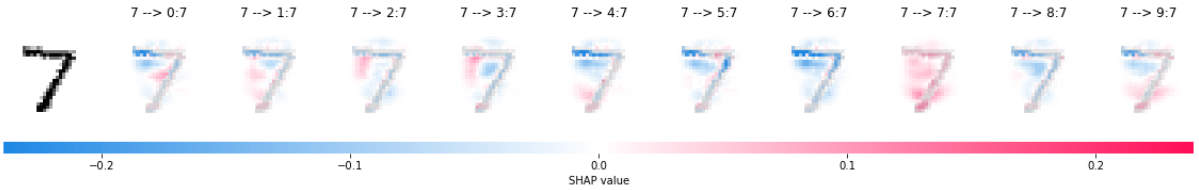
Εικόνα 70. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης πέντε (5) με το Deep SHAP



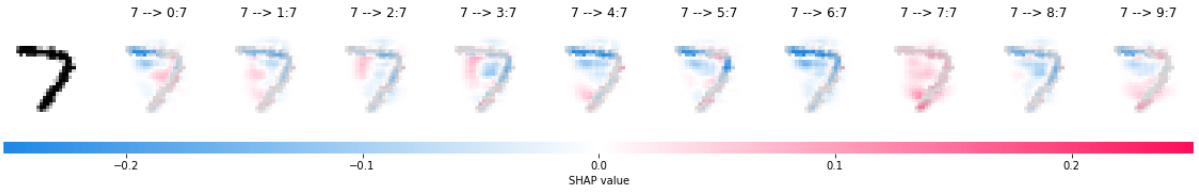
Εικόνα 71. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης έξη (6) με το Deep SHAP



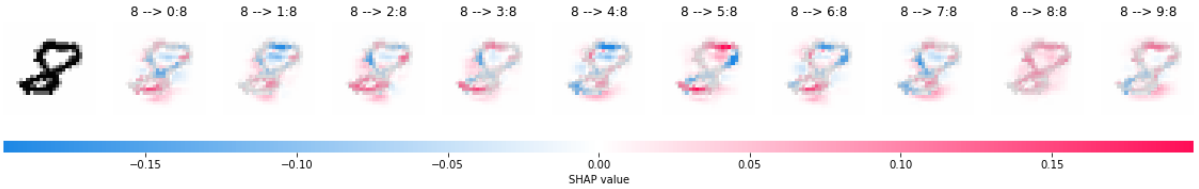
Εικόνα 72. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης έξη (6) με το Deep SHAP



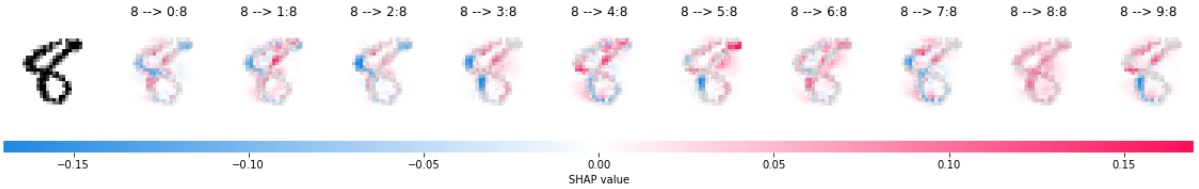
Εικόνα 73. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης επτά (7) με το Deep SHAP



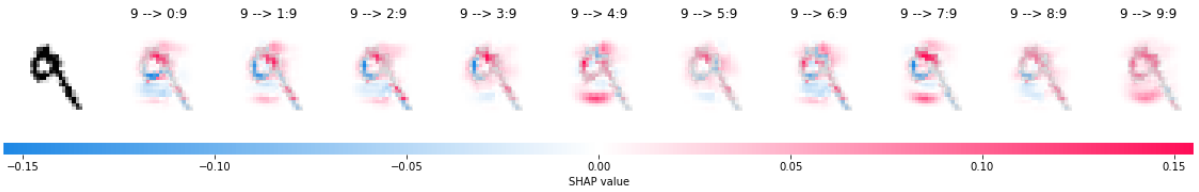
Εικόνα 74. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης επτά (7) με το Deep SHAP



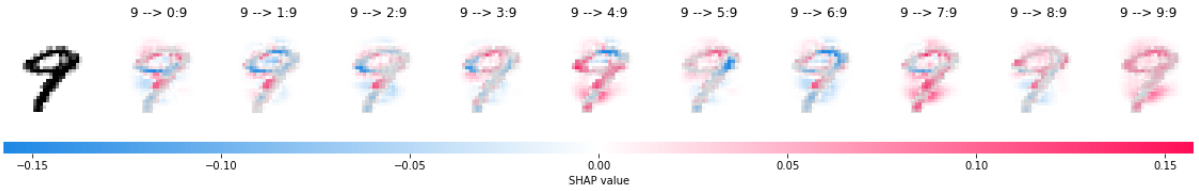
Εικόνα 75. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης οκτώ (8) με το Deep SHAP



Εικόνα 76. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης οκτώ (8) με το Deep SHAP

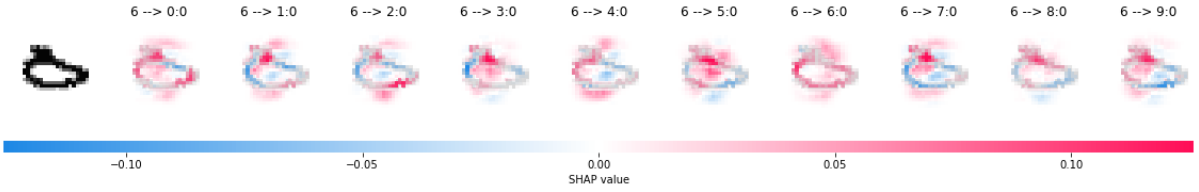


Εικόνα 77. Πρώτο δείγμα σωστής πρόβλεψης της κλάσης εννέα (9) με το Deep SHAP

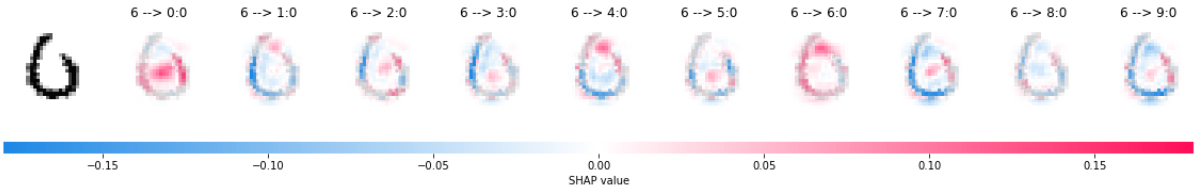


Εικόνα 78. Δεύτερο δείγμα σωστής πρόβλεψης της κλάσης εννέα (9) με το Deep SHAP

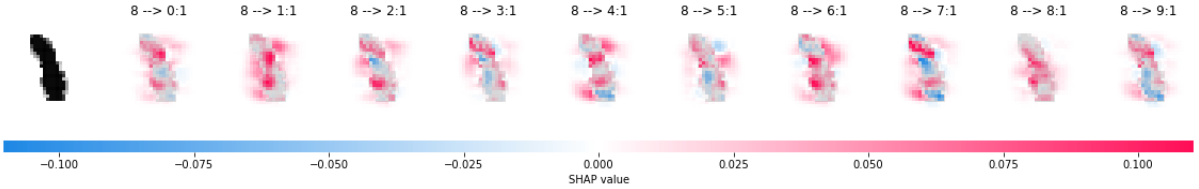
4.2.2 Λανθασμένες προβλέψεις



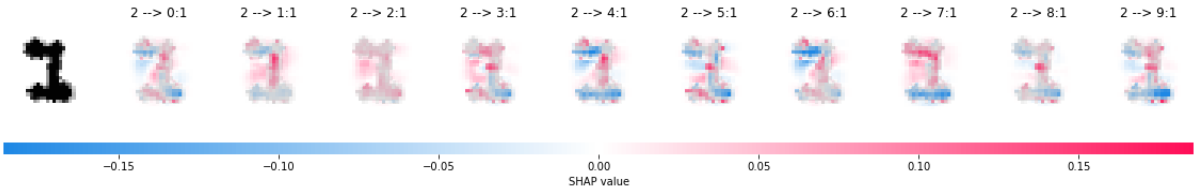
Εικόνα 79. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης μηδέν (0) με το Deep SHAP



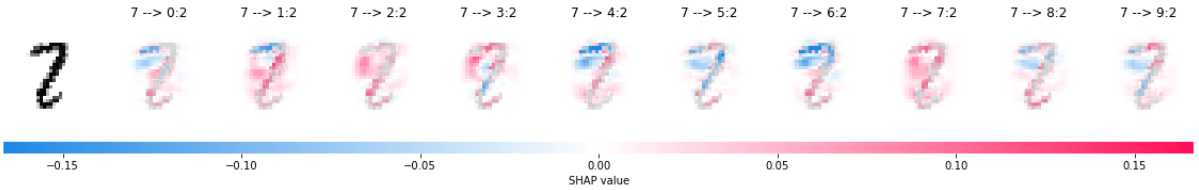
Εικόνα 80. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης μηδέν (0) με το Deep SHAP



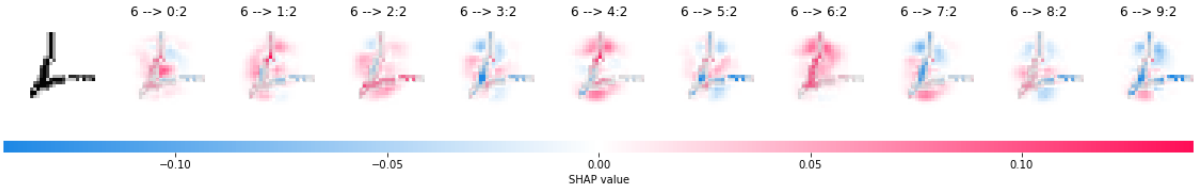
Εικόνα 81. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης ένα (1) με το Deep SHAP



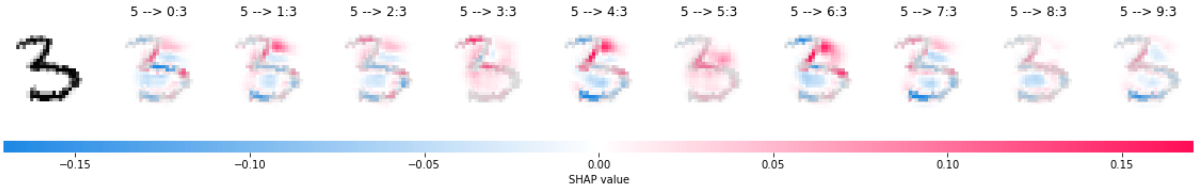
Εικόνα 82. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης ένα (1) με το Deep SHAP



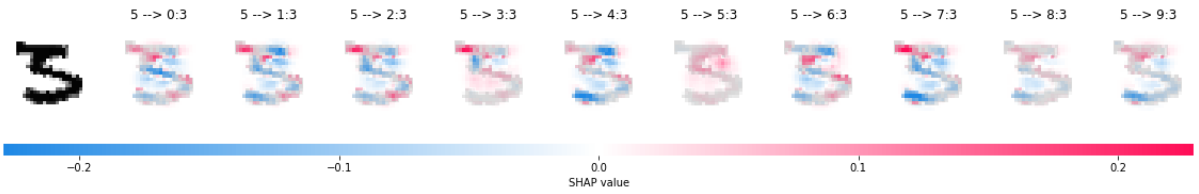
Εικόνα 83. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης δύο (2) με το Deep SHAP



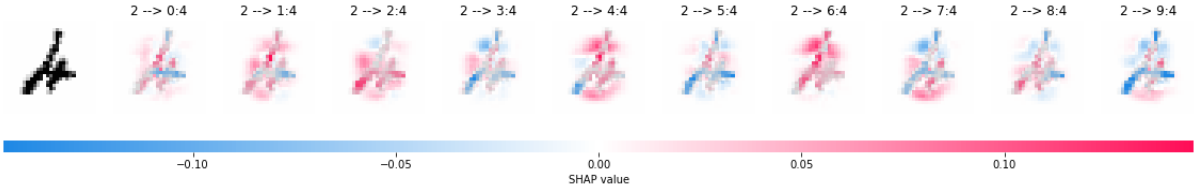
Εικόνα 84. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης δύο (2) με το Deep SHAP



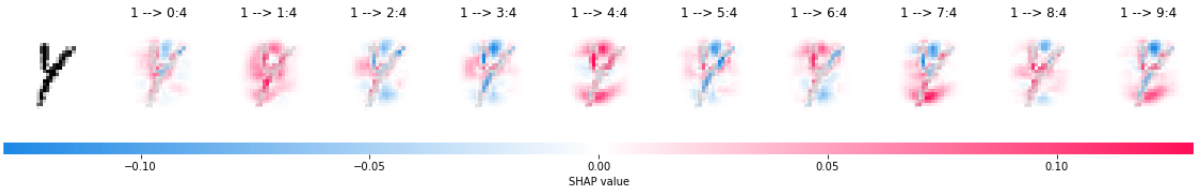
Εικόνα 85. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης τρία (0) με το Deep SHAP



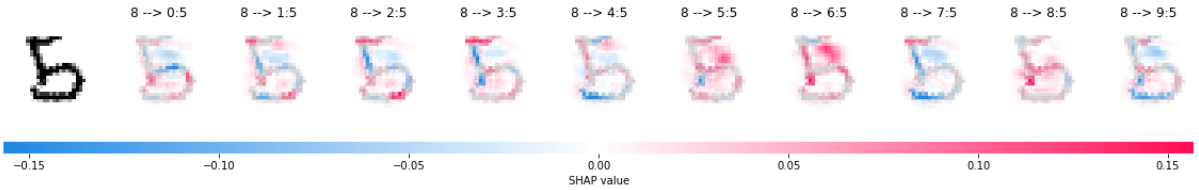
Εικόνα 86. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης τρία (3) με το Deep SHAP



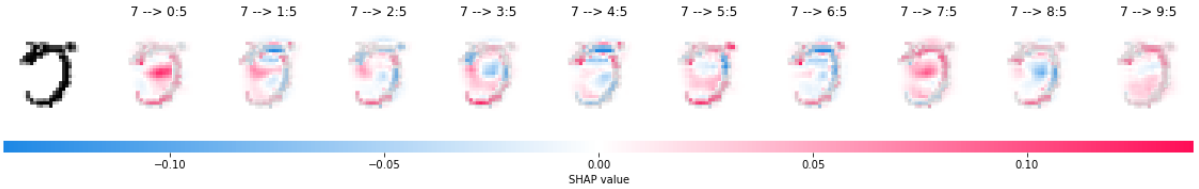
Εικόνα 87. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης τέσσερα (4) με το Deep SHAP



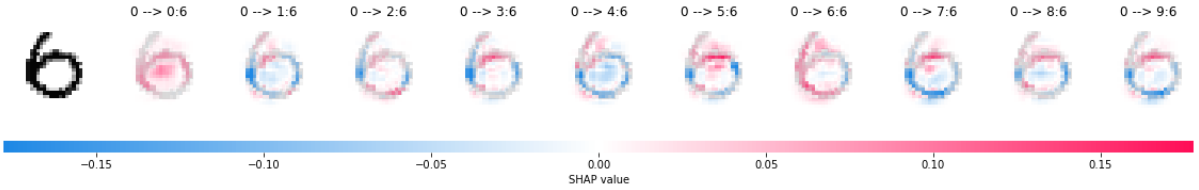
Εικόνα 88. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης τέσσερα (4) με το Deep SHAP



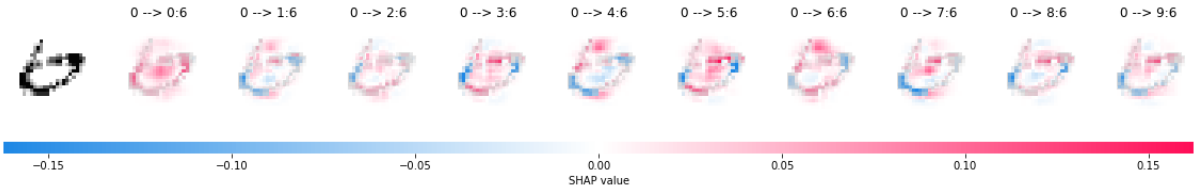
Εικόνα 89. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης πέντε (5) με το Deep SHAP



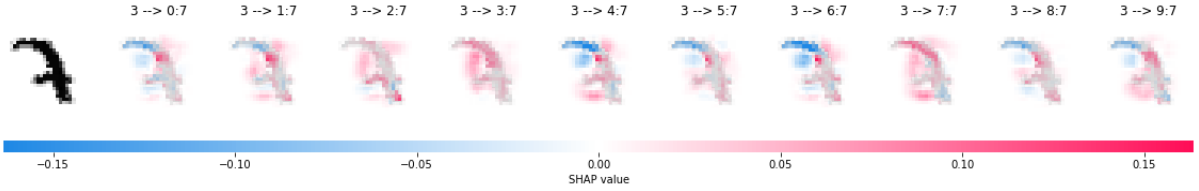
Εικόνα 90. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης πέντε (5) με το Deep SHAP



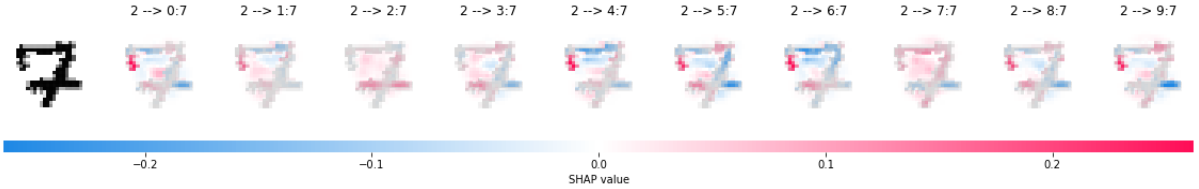
Εικόνα 91. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης έξη (6) με το Deep SHAP



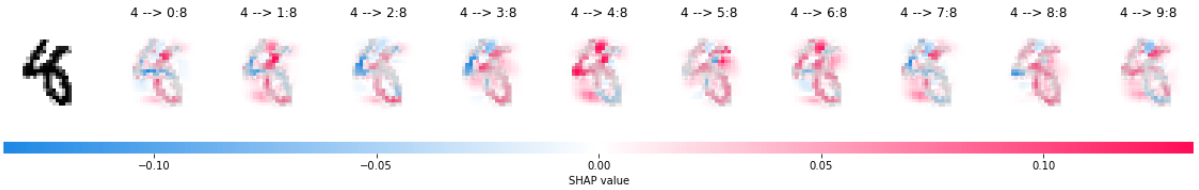
Εικόνα 92. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης έξη (6) με το Deep SHAP



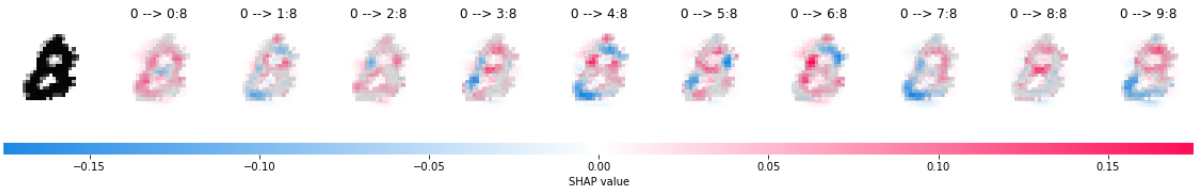
Εικόνα 93. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης επτά (7) με το Deep SHAP



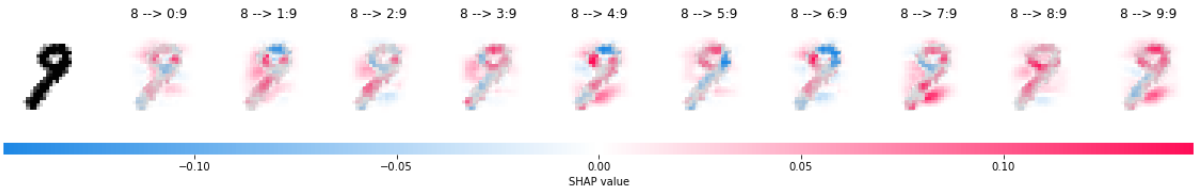
Εικόνα 94. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης επτά (7) με το Deep SHAP



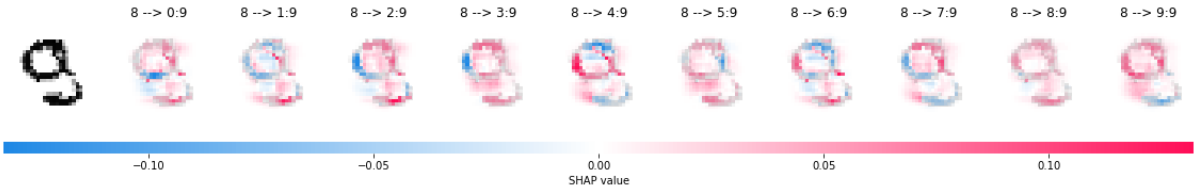
Εικόνα 95. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης οκτώ (8) με το Deep SHAP



Εικόνα 96. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης οκτώ (8) με το Deep SHAP



Εικόνα 97. Πρώτο δείγμα λανθασμένης πρόβλεψης της κλάσης εννέα (9) με το Deep SHAP



Εικόνα 98. Δεύτερο δείγμα λανθασμένης πρόβλεψης της κλάσης εννέα (9) με το Deep SHAP

5 Συμπεράσματα και μελλοντικές επεκτάσεις

Μέσω της παρούσας διπλωματικής εργασίας παρατηρήθηκε ότι ο κλάδος της επεξηγήσιμης τεχνητής νοημοσύνης σε λίγα χρόνια θα είναι πολύ σημαντικός και η εδραίωση καλύτερων μεθόδων με χαμηλό υπολογιστικό κόστος θα είναι όλο και πιο αναγκαίοι. Μέσω της εξήγησης των αποφάσεων των διαφανών μοντέλων θα μπορέσουν μηχανικοι, ερευνητές και χρήστες να εμπιστευτούν και να κατανοήσουν τα μοντέλα αυτα. Όσο για τους μηχανικούς και ερευνητές μπορεί μέσα από την εξήγηση του μοντέλου να παρθούν διάφορες πληροφορίες για το ίδιο το μοντέλο και τα δεδομένα που έχουν συλλεχθεί, όπως την πιθανή μορφή δεδομένων εκπαίδευσης που έχουν χρησιμοποιηθεί, πιθανά biases στα δεδομένα, πιθανά λάθη στην προεπεξεργασία των δεδομένων που θα έχουν δημιουργηθεί κατα την υλοποίηση ενός συστήματος και τα χαρακτηριστικά που έχουν πολύ σημαντικό ρόλο για τις προβλέψεις του μοντέλου.

Από την ενασχόληση στο πειραματικό μέρος, μπορούμε να πούμε ότι ο τρόπος με τον οποίο γίνεται κάθε φορά η προεπεξεργασία για τα δεδομένα έχει τον σημαντικότερο ρόλο. Διάφορα λάθη στην προεπεξεργασία μπορεί να κοστίσουν στην κατανόηση των αποτελεσμάτων του νευρωνικού δικτύου. Όμως πιθανά λάθη μπορούν να κατανοηθούν με την βοήθεια των μεθόδων εξήγησης. Η προεπεξεργασία κάθε φορά θα πρέπει να ελέγχεται ξεχωριστά πριν χρησιμοποιηθεί και να ελέγχεται η πιθανή αλλοίωση των δεδομένων μετά την χρήση της. Στο παράδειγμα μας της μεθόδου LIME η προεπεξεργασία που είχε γίνει αποτελούνταν από τρία κομμάτια κάτι που θεωρείται ιδιαίτερα επικίνδυνο καθώς θα μπορούσε να έχει αλλοιώσει τα αποτελέσματα της εξήγησης του δικτύου. Όμως αναγνωρίζεται ότι η υλοποίηση έχει τέτοια μορφή ώστε να μην περιορίζει το μοντέλο και την μορφή των δεδομένων (model-agnostic). Ενώ από την άλλη πλευρά η μέθοδος Deep SHAP όσον αφορά την προεπεξεργασία ήταν πολύ πιο εύκολη και το μόνο σημείο που μπορεί κάποιος να αρχίζει να αμφισβητεί είναι στην βασική προεπεξεργασία.

Απο τα δεδομένα των μεθόδων LIME και Deep SHAP, που καταφέραμε να κατεβάσουμε, φαίνεται ότι το νευρωνικό δίκτυο που χρησιμοποιήθηκε κατάφερε να καταλάβει ορισμένα χαρακτηριστικά κάθε κλάσης. Όσο πιο ευανάγνωστα τα χαρακτηριστικά τόσο το καλύτερο για το μοντέλο ενώ σε αντίθετη περίπτωση εάν δεν είναι ευανάγνωστα προκαλείται μια σύγχυση στο μοντέλο και λαμβάνει λάθος αποφάσεις (αρκετές φορές είναι κάπως λογικές).

Στην περίπτωση των αποτελεσμάτων του LIME παρότι βλέπαμε ότι συνέχεια τα αποτελέσματα τα οποία καλύπτουν περισσότερη περιοχή μπορεί να είναι και ο λόγος που επιλέχθηκε η κατάλληλη κλάση, υπήρξαν και περιπτώσεις που κάτι τέτοιο δεν ισχύει, αυτό παρατηρήθηκε κυρίως στις λανθασμένες αποφάσεις όπου δημιουργούνταν μία σύγχυση. Επομένως χαρακτηριστικά τα οποία πιθανό είναι πολύ πιο έντονα και έχουν σημαντικό ρόλο στην πρόβλεψη, το LIME μπορεί να αναδείξει το έντονο χαρακτηριστικό αλλά δεν μπορεί να απεικονίζει πλήρως τη σημαντικότητα. Αναγνωρίζεται και η πιθανότητα η παραμετροποίηση που χρησιμοποιήθηκε να μην είναι η καλύτερη που μπορούσε να εφαρμοστεί, κυρίως για το πόσα super-pixel πρέπει να συμπεριλάβει ή το κατώτατο βάρος να μην επέτρεψε στην καλύτερη απεικόνιση άλλων χαρακτηριστικών. Παρά τις αδυναμίες που παρατηρήθηκαν το LIME είναι ένα εργαλείο που έδωσε αρκετά καλά αποτελέσματα και αρκετά αξιόπιστα για την εφαρμογή μας. Σαν εργαλείο θα μπορούσε να χρησιμοποιηθεί καλύτερα σε πιο δύσκολα δεδομένα για την αναγνώριση περισσότερων χαρακτηριστικών σε πολύχρωμες εικόνες και τα αποτελέσματα που θα μας έδινε θα ήταν εύκολο να διακριθούν.

Στην περίπτωση των αποτελεσμάτων του Deep SHAP τα αποτελέσματα που λαμβάνουμε είναι με μεγαλύτερη λεπτομέρεια προς τα χαρακτηριστικά για κάθε κλάση και μπορούμε να κατανοήσουμε καλύτερα τις περιοχές που συνεισφέρουν θετικά για το δίκτυο μας καθώς και τις αρνητικές συνεισφορές. Περιπτώσεις που δημιουργείται σύγχυση άρχισαν να γίνονται πιο κατανοητές

και μπορούμε να τις επαληθεύσουμε καλύτερα έναντι του LIME. Το αρνητικό το οποίο έχει αυτή η μέθοδος είναι ότι σε ποιό περίπλοκες εικόνες τα αποτελέσματα θα ήταν κάπως χαοτικά για τον χρήστη, επομένως ένας καλύτερος τρόπος απεικόνισης θα είναι σημαντικός σε άλλες εφαρμογές. Με την βοήθεια του Deep SHAP μπορούμε να κατανοήσουμε και τις περιοχές που χρειάζεται να επηρεάσουμε στα δεδομένα ώστε να αρχίζουν να κατατάσσονται στις κλάσεις που πρέπει να ανήκουν.

Σαν μελλοντικές επεκτάσεις θα ήταν καλό να δοκιμαστούν και άλλες μέθοδοι και τεχνικές και να συγκριθούν με τα αποτελέσματα μας. Ακόμη σημαντικό θα ήταν η παραπάνω δοκιμή της μεθόδου LIME με διαφορετικές παραμετροποιήσεις ώστε να δούμε τις διαφορές, ή ακόμα και δοκιμη με παρόμοιες παραμέτρους που χρησιμοποιήθηκαν αλλα τα δεδομένα να είναι σε μορφή RGB που χρειάζεται το LIME ώστε να μην περιέχονται πολλές μετατροπές ενδιάμεσα και να συγκριθούν τα αποτελέσματα με τα αποτελέσματα της έρευνας, ώστε να διαπιστωθεί πόσο πολύ επηρεάστηκαν οι αποφάσεις από την προεπεξεργασία. Επίσης θα μπορούσαμε να επαναλάβουμε το πείραμα με ποιο σύνθετα δεδομένα για να διαπιστώσουμε εάν το LIME και το Deep SHAP θα εμφανίζουν και πάλι κατανοητά αποτελέσματα ή χρειάζονται άλλοι μέθοδοι απεικόνισης όπως Heatmaps.

Βιβλιογραφία και Διαδικτυακές πηγές

[1] Miller, T. (2019, April 10). "But why?" Understanding explainable artificial intelligence. Crossroads, The ACM Magazine for Students, Volume 25, Issue 3, pp 20–25. https://doi.org/10.1145/3313107

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, Volume 58, 2020, Pages 82-115. https://doi.org/10.1016/j.inffus.2019.12.012

[3] Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, Peter M. Atkinson, Explainable artificial intelligence: an analytical review, WIREs Data Mining and Knowledge Discovery, Volume 11, 2021, Issue 5, e1424, https://doi.org/10.1002/widm.1424

[4] Rai, A. Explainable AI: from black box to glass box. J. of the Acad. Mark. Sci. 48, 137–141 (2020). https://doi.org/10.1007/s11747-019-00710-5

[5] Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W. (2022). Explainable AI Methods - A Brief Overview. In: Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, KR., Samek, W. (eds) xxAI - Beyond Explainable AI. xxAI 2020. Lecture Notes in Computer Science(), vol 13200. Springer, Cham. https://doi.org/10.1007/978-3-031-04083-2_2

[6] Elhassan Mohamed, Konstantinos Sirlantzis, Gareth Howells. A review of visualisation-as-explanation techniques for convolutional neural networks and their valuation. Displays, Volume 73, July 2022, 102239.https://doi.org/10.1016/j.displa.2022.102239

[7] Molnar, C. Interpretable Machine Learning A Guide for Making Black Box Models Interpretable. Διαθέσιμο στο : https://books.google.gr/books?hl=el&lr=&id=jBm3DwAAQBAJ&oi=fnd&pg=PP1&ots=EgzTUpGIVY&sig=6YqYJtzo40OKpDM1BS1dV5sbZMc&redir_esc=y#v=onepage&q&f=false

[8] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K. -R. Müller. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. in Proceedings of the IEEE, vol. 109, no. 3, pp. 247-278, March 2021, DOI: 10.1109/JPROC.2021.3060483.

[9] Katarzyna Borys, Yasmin Alyssa Schmitt, Meike Nauta, Christin Seifert, Nicole Krämer, Christoph M. Friedrich, Felix Nensa. Explainable AI in medical imaging: An overview for clinical practitioners – Saliency-based XAI approaches. European Journal of Radiology, Volume 162, May 2023, 110787. https://doi.org/10.1016/j.ejrad.2023.110787

[10] Duval, Alexandre. (2019). Explainable Artificial Intelligence (XAI). DOI: 10.13140/RG.2.2.24722.09929.

[11] Ioannis Kakogeorgiou, Konstantinos Karantzalos. Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing. International Journal of Applied Earth Observation and Geoinformation, Volume 103, 1 December 2021, 102520. https://doi.org/10.1016/j.jag.2021.102520

[12] Shin-nosuke Ishikawa, Masato Todo, Masato Taki, Yasunobu Uchiyama, Kazunari Matsunaga, Peihsuan Lin, Taiki Ogihara, Masao Yasui. Example-based explainable AI and its application for remote sensing image classification. International Journal of Applied Earth Observation and Geoinformation, Volume 118, April 2023, 103215. https://doi.org/10.1016/j.jag.2023.103215s

[13] Bas H.M. van der Velden, Hugo J. Kuijf, Kenneth G.A. Gilhuijs, Max A. Viergever. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. Medical Image Analysis, Volume 79, July 2022, 102470. https://doi.org/10.1016/j.media.2022.102470

[14] Sajid Nazir, Diane M. Dickson, Muhammad Usman Akram. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. Computers in Biology and Medicine, Volume 156, April 2023, 106668. https://doi.org/10.1016/j.compbiomed.2023.106668

[15] Papastratis, Ilias (2021, March 4). Introduction to Explainable Artificial Intelligence (XAI). AI SUMMER. https://theaisummer.com/xai/

[16] Ancona, M., Ceolini, E., Öztireli, C., Gross, M. (2019). Gradient-Based Attribution Methods. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L., Müller, KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science(), vol 11700. Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_9

[17] Fong, R., Vedaldi, A. (2019). Explanations for Attributing Deep Neural Network Predictions. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L., Müller, KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science(), vol 11700. Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_8

[18] Bargal, S.A. et al. (2022). Beyond the Visual Analysis of Deep Model Saliency. In: Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, KR., Samek, W. (eds) xxAI - Beyond Explainable AI. xxAI 2020. Lecture Notes in Computer Science(), vol 13200. Springer, Cham. https://doi.org/10.1007/978-3-031-04083-2_13

[19] Weller, A. (2019). Transparency: Motivations and Challenges. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L., Müller, KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science(), vol 11700. Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_2

[20] Bauer, K., Hinz, O., van der Aalst, W. et al. Expl(AI)n It to Me – Explainable AI and Information Systems Research. Bus Inf Syst Eng 63, 79–82 (2021). https://doi.org/10.1007/s12599-021-00683-2

[21] Benedikt Leichtmann, Christina Humer, Andreas Hinterreiter, Marc Streit, Martina Mara. Effects of Explainable Artificial Intelligence on trust and human behavior in a high-risk decision task. Computers in Human Behavior, Volume 139, 2023, 107539. https://doi.org/10.1016/j.chb.2022.107539.

[22] Buijsman, S. Defining Explanation and Explanatory Depth in XAI. Minds & Machines 32, 563–584 (2022). https://doi.org/10.1007/s11023-022-09607-9

[23] Rajpal, A., Sehra, K., Bagri, R. et al. XAI-FR: Explainable AI-Based Face Recognition Using Deep Neural Networks. Wireless Pers Commun 129, 663–680 (2023). https://doi.org/10.1007/s11277-022-10127-z

[24] Holzinger, A., Kieseberg, P., Weippl, E., Tjoa, A.M. (2018). Current Advances, Trends and Challenges of Machine Learning and Knowledge Extraction: From Machine Learning to Explainable AI. In: Holzinger, A., Kieseberg, P., Tjoa, A., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2018. Lecture Notes in Computer Science(), vol 11015. Springer, Cham. https://doi.org/10.1007/978-3-319-99740-7_1

[25] Markus Langer, Daniel Oster, Timo Speith, Holger Hermanns, Lena Kästner, Eva Schmidt, Andreas Sesing, Kevin Baum. What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. Artificial Intelligence, Volume 296, July 2021, 103473.https://doi.org/10.1016/j.artint.2021.103473

[26] Paolo Giudici, Emanuela Raffinetti. Shapley-Lorenz eXplainable Artificial Intelligence. Expert Systems with Applications, Volume 167, 1 April 2021, 114104. https://doi.org/10.1016/j.eswa.2020.114104

[27] M.Z. Naser. An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating causality, forced goodness, and the false perception of inference. Automation in Construction, Volume 129, September 2021, 103821. https://doi.org/10.1016/j.autcon.2021.103821

[28] Xiaoge Zhang, Felix T.S. Chan, Sankaran Mahadevan. Explainable machine learning in image classification models: An uncertainty quantification perspective. Knowledge-Based Systems, Volume 243, 11 May 2022, 108418. https://doi.org/10.1016/j.knosys.2022.108418

[29] Richard Dazeley, Peter Vamplew, Cameron Foale, Charlotte Young, Sunil Aryal, Francisco Cruz.Levels of explainable artificial intelligence for human-aligned conversational explanations. Artificial Intelligence, Volume 299, October 2021, 103525. https://doi.org/10.1016/j.artint.2021.103525

[30] Zednik, C., Boelsen, H. Scientific Exploration and Explainable Artificial Intelligence. Minds & Machines 32, 219–239 (2022). https://doi.org/10.1007/s11023-021-09583-6

[31] Yang, Yuting & Mei, Gang & Piccialli, Francesco. (2021). Explainable Deep Learning Models on the Diagnosis of Pneumonia. DOI: 10.1109/CHASE52844.2021.00032.

[32] Mukhtorov, Doniyrojon & Madinakhon, Rakhmonova & Muksimova, Shakhnoza & Cho, Young-Im. (2023). Endoscopic Image Classification Based on Explainable Deep Learning. Sensors. 23. 3176. DOI: 10.3390/s23063176.

[33] Singh, Amitojdeep & Sengupta, Sourya & Lakshminarayanan, Vasudevan. (2020). Explainable Deep Learning Models in Medical Image Analysis. Journal of Imaging. 6. 52. DOI: 10.3390/jimaging6060052.

[34] Explainable Artificial Intelligence (XAI) (Archived). DAPRA. Διαθέσιμο στο: https://www.darpa.mil/program/explainable-artificial-intelligence

[35]What is PyTorch?. NVIDIA. Διαθέσιμο στο: https://www.nvidia.com/en-us/glossary/data-science/pytorch/

[36]How to use TensorBoard with PyTorch. NVIDIA. Διαθέσιμο στο: https://pytorch.org/tutorials/recipes/recipes/tensorboard_with_pytorch.html

[37]MNIST database. In Wikipedia. Διαθέσιμο στο: https://en.wikipedia.org/wiki/MNIST_database

[38] Datasets. NVIDIA. Διαθέσιμο στο: https://pytorch.org/vision/main/datasets.html

[39]What is MNIST? And why is it important?. Medium. Διαθέσιμο στο: https://selectstar-ai.medium.com/what-is-mnist-and-why-is-it-important-e9a269edbad5

[40]MNIST. NVIDIA. Διαθέσιμο στο: https://pytorch.org/vision/main/generated/torchvision.datasets.MNIST.html#torchvision.datasets.MNIST

[41] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv. https://doi.org/10.48550/arXiv.1602.04938

[42] Avanti Shrikumar, Peyton Greenside, Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. arXiv. https://doi.org/10.48550/arXiv.1704.02685

[43] Peng Yu, Chao Xu, Albert Bifet, Jesse Read. Linear TreeShap. arXiv. https://doi.org/10.48550/arXiv.2209.08192

[44]MNIST Handwritten Digit Recognition in PyTorch. Nextjournal. Διαθέσιμο στο: https://nextjournal.com/gkoehler/pytorch-mnist

[45]PyTorch: Training your first Convolutional Neural Network (CNN). pyimagesearch. Διαθέσιμο στο: https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/

[46] Saranya A., Subhashini R.. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. Decision Analytics Journal, Volume 7, June 2023, 100230. https://doi.org/10.1016/j.dajour.2023.100230

[47] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad, Jose M. Alonso-Moral, Roberto Confalonieri, Riccardo Guidotti, Javier Del Ser, Natalia Díaz-Rodríguez, Francisco Herrera. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. Information Fusion, Volume 99, November 2023, 101805. https://doi.org/10.1016/j.inffus.2023.101805

[48] Amin Nayebi, Sindhu Tipirneni, Brandon Foreman, Chandan K. Reddy, Vignesh Subbian. An Empirical Comparison of Explainable Artificial Intelligence Methods for Clinical Data: A Case Study on Traumatic Brain Injury. arXiv. https://doi.org/10.48550/arXiv.2208.06717

[49] Kamakshi, V., Krishnan, N.C.. Explainable Image Classification: The Journey So Far and the Road Ahead. AI 2023, 4, 620-651. https://doi.org/10.3390/ai4030033