

## Computational Intelligence CEID1060

---

### Assignment Report

# Πρόβλεψη **Alzheimer's** με Χρήση Νευρωνικών Δικτύων

---

Advisor Instructor: D. Koutsomitropoulos, S. Likothanasis

AM	Surname	Name	Semester
1090034	Nasieli	Vasiliki	10 Spring Semester <sup>o</sup>

<https://github.com/vicdo18/Computational-Intelligence>

## Contents

<b>Chapter 1:</b>	<b>A1. Προεπεξεργασία και Προετοιμασία δεδομένων</b>	<b>3</b>
1.a	Alzheimer's Disease Dataset . . . . .	3
1.b	a.Κωδικοποίηση και προεπεξεργασία δεδομένων . . . . .	3
1.b.i	b. Διασταυρούμενη Επικύρωση (cross-validation) . . . . .	3
<b>Chapter 2:</b>	<b>A2. Επιλογή αρχιτεκτονικής</b>	<b>4</b>
2.a	α. . . . .	4
2.b	β. . . . .	4
2.c	γ. . . . .	4
2.d	δ. . . . .	5
2.e	ε. . . . .	5
2.e.i	Συμπεράσματα Πειραμάτων . . . . .	6
2.f	στ. Κριτήριο τερματισμού . . . . .	7
<b>Chapter 3:</b>	<b>A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής</b>	<b>8</b>
3.a	Γιατί $m < 1$ ; . . . . .	9
<b>Chapter 4:</b>	<b>A4. Ομαλοποίηση</b>	<b>9</b>
<b>Chapter 5:</b>	<b>A5. Βαθύ Νευρωνικό Δίκτυο</b>	<b>11</b>
5.a	Μοτίβα στην Αρχιτεκτονική . . . . .	12



## Chapter 1: A1. Προεπεξεργασία και Προετοιμασία δεδομένων

### 1.a Alzheimer's Disease Dataset

#### 1.b a.Κωδικοποίηση και προεπεξεργασία δεδομένων

Για τον καλύτερο δυνατό χειρισμό του συγκεκριμένου **dataset**, χρειάστηκε να εκτελεστούν ορισμένα βήματα προεπεξεργασίας. Συγκεκριμένα για το πρώτο ερώτημα, αφού φορτωθεί το αρχείο **csv**, "κόβουμε" τις στήλες 'PatientID', 'DoctorInCharge' καθώς δεν συνεισφέρουν στην αργότερη δημιουργία του μοντέλου ή σε κάποια άλλη διαδικασία.

Αμέσως μετά, γίνεται έλεγχος ύπαρξης μηδενικών (**null**) τιμών και διαπιστώνεται πως δεν υπάρχουν.

Για διευκόλυνση της διαδικασίας, τοποθετούμε τα χαρακτηριστικά/ονόματα στηλών σε λίστες που υποδεικνύουν τον τύπο δεδομένων που περιέχουν:

**continuous\_cols:** Κυρίως αριθμητικές τιμές, που ανήκουν σε συγκεκριμένο διάστημα τιμών.

**categorical\_cols:** Κατηγορικές τιμές, που διακρίνονται σε επίπεδα. Και πάλι αριθμητικές τιμές. Η χρήση αυτής της λίστας δεν είναι απαραίτητη στο σημείο που βρισκόμαστε.

**binary\_cols:** Λίστα δυαδικών τιμών.

Μετά την δημιουργία των παραπάνω λιστών, απομακρύνουμε τις μεταβλητές **target**, οι οποίες αποτελούν την στήλη 'Diagnosis' του **dataset**.

Θα περάσουμε τα δεδομένα, συγκεκριμένα τις κατηγορικές τιμές από έναν **MinMax scaler**, για την κανονικοποίηση του εύρους, της κατανομής και του μεγέθους των χαρακτηριστικών, μειώνοντας πιθανές προκαταλήψεις και ασυνέπειες που μπορεί να προκύψουν από διακυμάνσεις στις τιμές τους.

Τέλος, αποθηκεύουμε το νέο προεπεξεργασμένο **dataset (.csv)**.

#### 1.b.i b. Διασταυρούμενη Επικύρωση (**cross-validation**)



## Chapter 2: A2. Επιλογή αρχιτεκτονικής

### 2.a α.

Η **Cross Entropy loss**, μας βοηθάει να καταλάβουμε πόσο καλά ουσιαστικά το μοντέλο μας διαχωρίζει τις δύο κλάσεις του **target** ( **Alzheimer's[1]** vs **non Alzheimer's[0]**). Πιο αναλυτικά, χαμηλό **CE** σημαίνει πως το μοντέλο αποδίδει ικανοποιητικά και κάνει σωστές προβλέψεις. Θα προτιμήσουμε την συγκεκριμένη μετρική για εκπαίδευση καθώς είναι ιδανική για περιπτώσεις κατηγοριοποίησης και επιπλέον χειρίζεται πιθανές ανισορροπίες καλύτερα.

Το **MSE (Mean Squared Error)**, μετρά την μέση τετραγωνική διαφορά μεταξύ των προβλεπόμενων πιθανοτήτων και των πραγματικών **labels**.

Το **Accuracy**, ως μετρική αναπαριστά το ποσοστό των σωστών προβλέψεων του μοντέλου.

### 2.b β.

Το συγκεκριμένο πρόβλημα εντάσσεται στην κατηγορία της δυαδικής ταξινόμησης (**binary classification**), καθώς έχουμε δύο διαφορετικές κλάσεις: **Alzheimer's[1]** και **non Alzheimer's[0]**. Επομένως μας αρκεί ένας νευρώνας στο επίπεδο εξόδου. Εάν είχαμε πάνω από **2** κλάσεις θα χρειαζόμασταν έναν νευρώνα ανά κλάση. Στην περίπτωση μας, οι πιθανότητες είναι αμοιβαία αποκλειόμενες.

### 2.c γ.

Για το πρόβλημα της δυαδικής ταξινόμησης (**Alzheimer vs. non-Alzheimer**), η επιλογή της **ReLU (Rectified Linear Unit)** ως συνάρτησης ενεργοποίησης του κρυφού επιπέδου είναι η πιο κατάλληλη για τους κρυφούς κόμβους, με **Tanh** και **SiLU** να είναι εναλλακτικές ανάλογα με την περίπτωση.

Τύπος **ReLU**:  $f(x)=\max(0,x)$

Στα πλεονεκτήματα χρήσης της **ReLU** εντάσσεται η υψηλή της υπολογιστική απόδοση καθώς χαρακτηρίζεται από μη γραμμικότητα χωρίς πολύπλοκους υπολογισμούς. Επιπλέον, γίνεται αποτελεσματικά, γρήγορα η εκπαίδευση και αποφεύγεται το φαινόμενο **vanishing gradient** όπου σε μεγάλα **x**, η παράγωγος είναι **1**.

Εάν εφαρμόσουμε την **ReLU** στο κρυφό επίπεδο έχουμε:

```
=== Average Cross-Validation Metrics ===
Mean accuracy: 0.8325 (±0.0109)
Mean precision: 0.8077 (±0.0534)
Mean recall: 0.7005 (±0.0490)
Mean f1: 0.7471 (±0.0120)
Mean roc_auc: 0.8961 (±0.0144)
Mean mse: 0.1675 (±0.0109)
```



Στην περίπτωση χρήσης της **Tanh** (Υπερβολική Εφαπτομένη), απαιτώνται εκθετικοί υπολογισμοί άρα καθίσταται πιο αργή:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Η **SiLU (Sigmoid-Weighted Linear Unit)**, γενικά αποδίδει καλά σε βαθιά νευρωνικά δίκτυα και συνδυάζει τα καλά της **ReLU** και **Sigmoid**, είναι όμως για την περίπτωσή μας πιο περίπλοκη και θα ήταν χρήσιμη σε περιπτώσεις όπου δεν υπάρχει υπολογιστικό περιθώριο.

$$f(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}$$

Εάν εφαρμόσουμε την **SiLU** στο κρυφό επίπεδο έχουμε:

```
=== Average Cross-Validation Metrics ===  
Mean accuracy: 0.8342 (±0.0162)  
Mean precision: 0.7915 (±0.0573)  
Mean recall: 0.7350 (±0.0611)  
Mean f1: 0.7579 (±0.0192)  
Mean roc_auc: 0.9006 (±0.0158)  
Mean mse: 0.1658 (±0.0162)
```

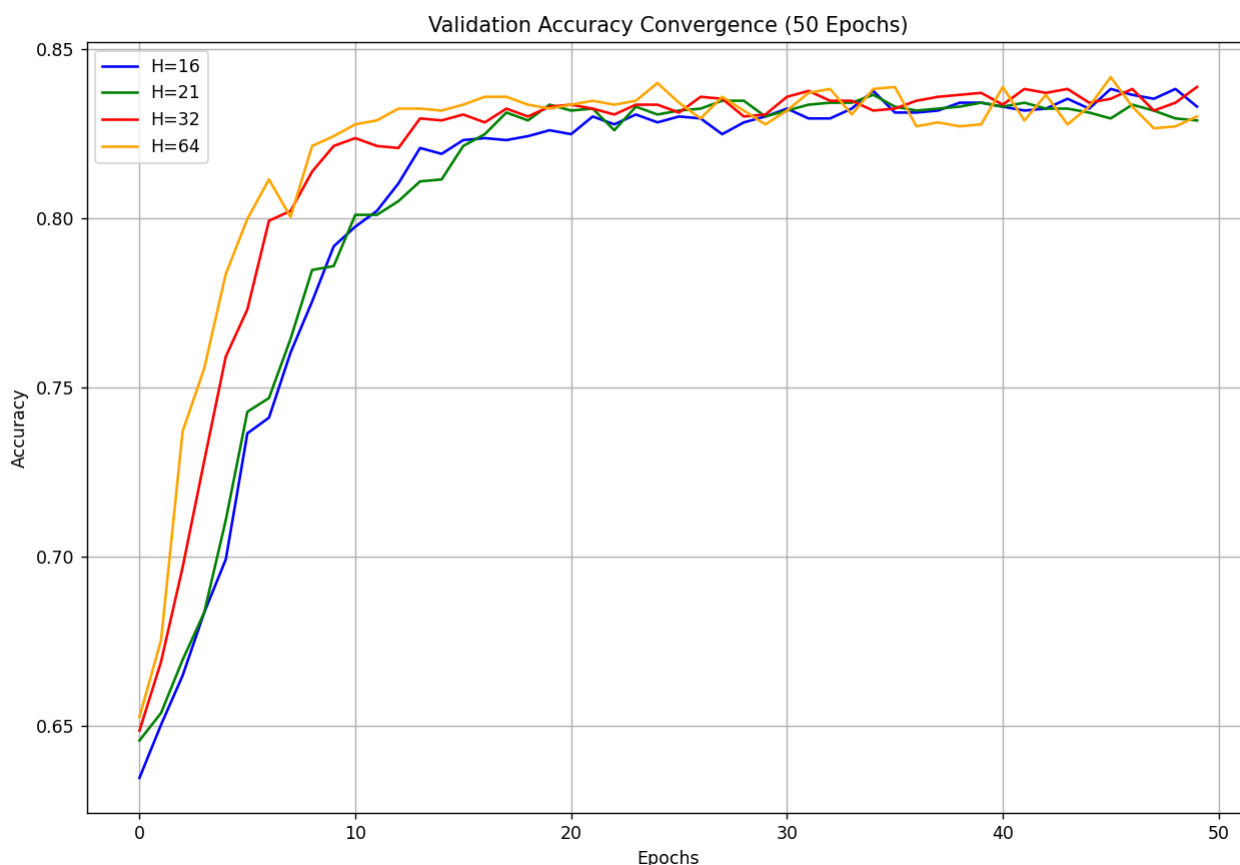
## 2.d δ.

Για το επίπεδο εξόδου, η καλύτερη δυνατή προσέγγιση είναι να χρησιμοποιήσουμε την σιγμοειδή συνάρτηση, κυρίως επειδή το πρόβλημα ταξινόμησης είναι δυαδικό.

## 2.e ε.

Αριθμός νευρώνων στο κρυφό επίπεδο	CE loss	MSE	Acc
$H_l = I/2(16)$	0.386	0.117	$0.837 \pm 0.016$
$H_l = 2I/3$	0.389	0.118	$0.834 \pm 0.014$
$H_l = I(32)$	0.396	0.12	$0.837 \pm 0.019$
$H_l = 2I(64)$	0.399	0.122	$0.831 \pm 0.020$

Table 1: Performance metrics for various hidden layer neuron counts



## 2.e.i Συμπεράσματα Πειραμάτων

Συμπερασματικά, η μελέτη διαφορετικών αρχιτεκτονικών ( $H = 1/2, 2/3, 1, 2/1$ ) έδειξε ότι το μοντέλο μας είχε την βέλτιστη ισορροπημένη απόδοση στους **16** νευρώνες κρυφού επιπέδου. Συγκεκριμένα στην περίπτωση αυτή ( $H = 16$ ) έχουμε την χαμηλότερη τιμή κόστους (**CE Loss**) και **MSE**. Επιπλέον, για **50 Epochs**, έχουμε την υψηλότερη ταχύτητα σύγκλισης καθώς και την υψηλότερη ακρίβεια .

Όσο το δίκτυό μας μεγαλώνει, για  $H=64$ , έχουμε οριακά χειρότερες μετρήσεις: αύξηση του **CE Loss** (**+3.4%** σε σχέση με  $H=16$ ) και απαίτηση μεγαλύτερου αριθμού εποχών για σύγκλιση (περίπου **50%**). Επιπλέον εμφανίζεται υψηλότερη διακύμανση στις τιμές Ακρίβειας. Επομένως για το συγκεκριμένο πρόβλημα με τις διαθέσιμες έως τώρα παραμέτρους και με βάση την αρχή του **Occam's razor**, μικρότερα και πιο απλά δίκτυα είναι αποδοτικότερα. Προτείνεται:  $H \in [1/2, 1]$  με προτίμηση στο  $H = 1/2$ .

Όσον αφορά την συνάρτηση κόστους, η **Binary Cross Entropy** επιλέχθηκε ως η βέλτιστη καθώς είναι η κατάλληλη για δυαδική ταξινόμηση αφού εκτιμά πιθανότητες και επιπλέον επιβάλλει αυστηρές ποινές σε λανθασμένες προβλέψεις. Αποτελεί καλύτερη επιλογή από **MSE** για **classification tasks** καθώς μπορούμε να αποφύγουμε **saturation** σε **sigmoid** και επιπλέον έχουμε ευαισθησία σε ασυμμετρικές κλάσεις.

Με την χρήση της **ReLU** ως συνάρτηση ενεργοποίησης κρυφών κόμβων, βλέπουμε μεγαλύτερη αποδοτικότητα όπως είχαμε προβλέψει. Στο πείραμα έχουμε ταχύτερη σύγκλιση σε σύγκριση με άλλες συναρτήσεις πχ, **leaky ReLU**, και δεν παρατηρούμε φαινόμενο "νεκρών νευρώνων".



## 2.f στ. Κριτήριο τερματισμού

Κοιτάζοντας τα αποτελέσματα των πειραμάτων μας, μπορούμε να δούμε ότι όλες οι αρχιτεκτονικές μοντέλων έφτασαν με φυσικό τρόπο στην κορυφαία τους απόδοση πολύ πριν ολοκληρωθούν και οι **50** εποχές. Για παράδειγμα, το μοντέλο **16** νευρώνων βρήκε τα βέλτιστα βάρη του στην εποχή **40**, ενώ το μεγαλύτερο μοντέλο **64** νευρώνων σταθεροποιήθηκε ακόμη νωρίτερα στην εποχή **30**. Αυτό το μοτίβο δείχνει ξεκάθαρα ότι η συνέχιση της εκπαίδευσης πέρα από αυτά τα σημεία θα σπαταλούσε απλώς υπολογιστικούς πόρους χωρίς να βελτιώνει τα αποτελέσματα. Χρησιμοποιήσαμε την τεχνική του πρόωρου σταματήματος (**early stopping**) με τις εξής παραμέτρους:

```
1 EarlyStopping(  
2     monitor='val_loss',  
3     patience=5,  
4     restore_best_weights=True  
5 )
```

Listing 1: Parameters

Με την παράμετρο **monitor**, δηλώνουμε παρακολούθηση της μεταβλητής **validation loss** αντί της ακρίβειας επειδή είναι πιο ευαίσθητη σε λεπτές αλλαγές στην απόδοση του μοντέλου. Με το **patience=5** ουσιαστικά δίνουμε στο μοντέλο αρκετό χρόνο για να ξεπεράσει πιθανώς προσωρινά πλατώματα απόδοσης (**plateaus**), ενώ παράλληλα αποτρέπει την περιττή εκτεταμένη εκπαίδευση. Θέτοντας το **restore\_best\_weights** ως **True**, επιτρέπουμε την επαναφορά των βέλτιστων βαρών, διασφαλίζοντας ότι διατηρούμε την ιδανική έκδοση του μοντέλου.

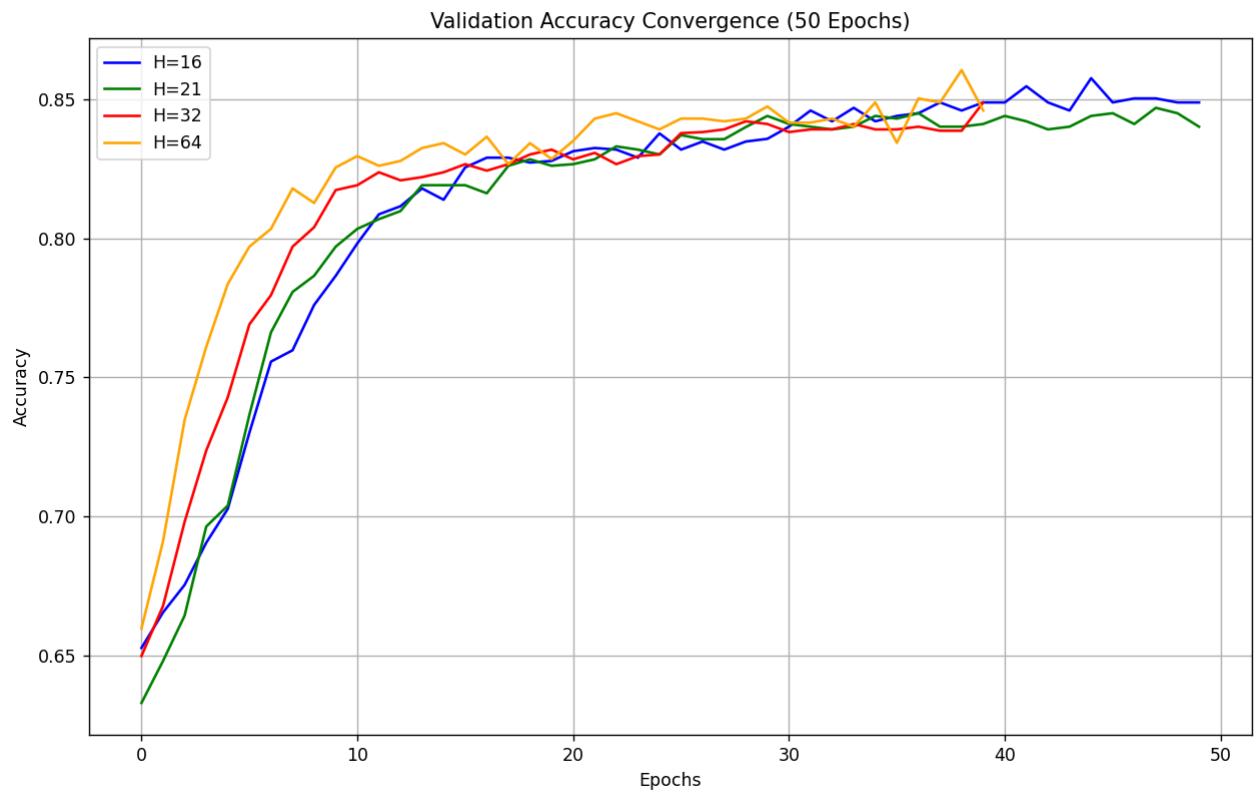
Σε σύγκριση με την εκπαίδευση για έναν καθορισμένο αριθμό εποχών, το πρόωρο σταμάτημα παρέχει τρία βασικά πλεονεκτήματα:

1. Αποτρέπει το μοντέλο από το να "ξεμάνει" καλά πρότυπα (όπως φαίνεται από τις σταθερές τιμές απώλειας **CE**).
2. Προσαρμόζεται αυτόματα στα μοναδικά χαρακτηριστικά κάθε πτυχής (**fold**) κατά τη διάρκεια του **cross validation**.
3. Μειώνει σημαντικά τον χρόνο εκπαίδευσης - στην περίπτωσή μας κατά περίπου **20-40%** ανάλογα με την αρχιτεκτονική.

Επιλέον, η τεχνική αυτή είναι ιδανική για το συγκεκριμένο πρόβλημα διάγνωσης διότι το μέγεθος του **dataset** είναι ελεγχόμενο, και επιπλέον χρησιμοποιούμε βελτιστοποιητή **Adam** ο οποίος συμπληρώνει ικανοποιητικά το πρόωρο σταμάτημα. Αυτό συμβαίνει επειδή οι προσαρμοστικοί ρυθμοί μάθησης αποτρέπουν τις πρόωρες διακοπές σε ψευδή πλατώματα, ενώ εξακολουθούν να ανιχνεύουν την πραγματική σύγκλιση. Αυτή η συμπεριφορά που μοιάζει με ορμή βοηθά τα μοντέλα να ξεπεράσουν προσωρινές "στάσεις", επιτρέποντας στην ανοχή των **5** εποχών που έχουμε ορίσει να λειτουργεί με συνέπεια σε όλες τις αρχιτεκτονικές. Αυτή η συνέργεια εξηγεί γιατί όλα τα μοντέλα μας πέτυχαν ισχυρή, σταθερή ακρίβεια (**0,83+**) ενώ σταμάτησαν την εκπαίδευση ακριβώς τη σωστή στιγμή για κάθε αρχιτεκτονική.



=== Performance Summary (50 Epochs) ===					
Hidden Units	Val Accuracy	MSE	CE Loss	Val F1	Val ROC AUC
16	0.837 ± 0.013	0.119565	0.388796	0.777	0.898 ± 0.016
21	0.834 ± 0.013	0.119338	0.38548	0.781	0.899 ± 0.017
32	0.834 ± 0.016	0.119543	0.388152	0.779	0.898 ± 0.017
64	0.832 ± 0.011	0.121327	0.393986	0.786	0.894 ± 0.016

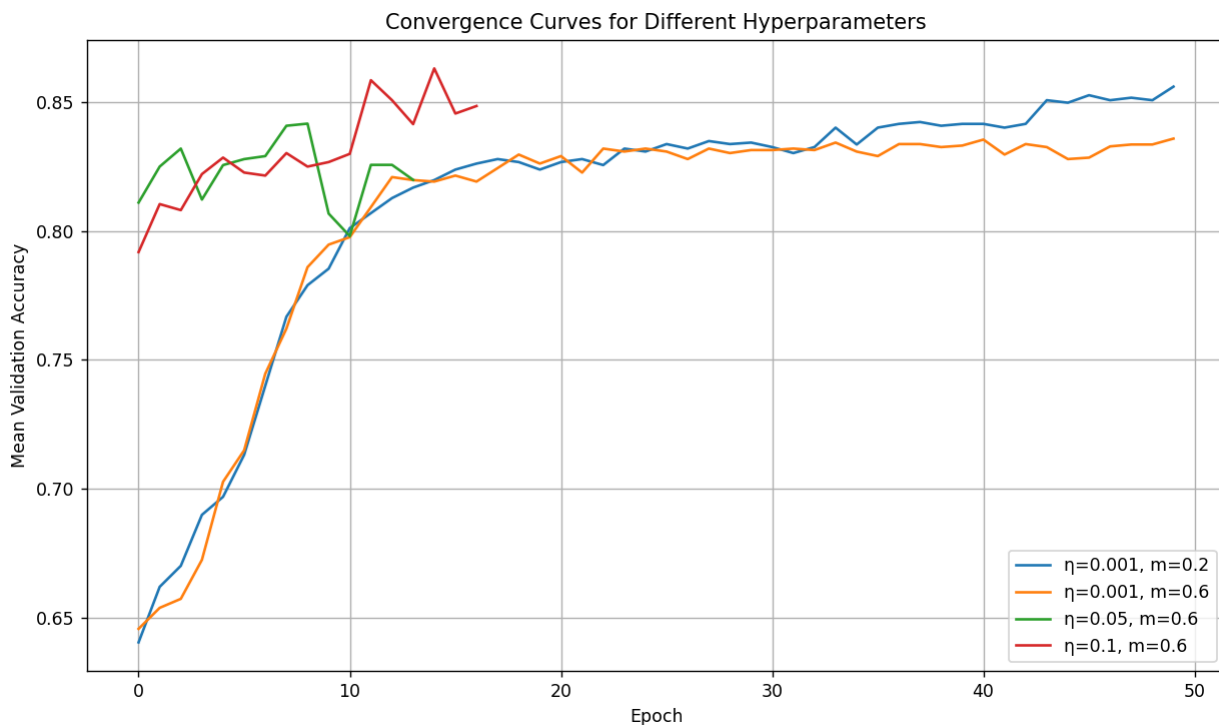


### Chapter 3: A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

$\eta$	m	CE loss	MSE	Acc
0.001	0.2	0.3896	0.1199	0.8383
0.001	0.6	0.3934	0.1209	0.8336
0.05	0.6	0.414	0.1279	0.8272
0.1	0.6	0.4311	0.1279	0.822

Table 2: Τιμές μεταβολών





Το παραπάνω γράφημα παρουσιάζει τη συμπεριφορά σύγκλισης διαφορετικών συνδυασμών των υπερ-παραμέτρων ρυθμού μάθησης  $\eta$  και ορμής  $m$  με βάση τη μέση ακρίβεια επικύρωσης σε **5** αναδιπλώσεις **cross validation**, κατά μήκους **50** εποχών.

Οι πιο απότομες καμπύλες στην αρχή (όπως η κόκκινη και η πράσινη) σημαίνουν ταχύτερη εκμάθηση. Αυτοί οι συνδυασμοί βοήθησαν το μοντέλο μας να επιτύχει υψηλότερη ακρίβεια γρηγορότερα.

Κάθε καμπύλη είναι:

Η μέση ακρίβεια επικύρωσης σε **5** αναδιπλώσεις.

Απεικονίζεται ανά εποχή για έναν δεδομένο συνδυασμό των  $(\eta, m)$ .

Το μοντέλο ξεκινά με ακρίβεια περίπου **0,65** και βελτιώνεται σε **0,85** στον άξονα  $Y$ , το οποίο αποτελεί τυπικό εύρος για δυαδική ταξινόμηση (όταν **0,5** = τυχαία εικασία). Παρατηρούμε πως την καλύτερη διαμόρφωση απόδοσης την έχει η πρώτη περίπτωση συνδυασμού παραμέτρων, αγγίζοντας περίπου το **85%** ακρίβειας.

### 3.a Γιατί $m < 1$ ;

Η ορμή ( $m$ ) συνήθως διατηρείται κάτω από το **1** επειδή, όταν  $m = 1$ , η ενημέρωση της ταχύτητας θα εξαρτιόταν εξ ολοκλήρου από την προηγούμενη ταχύτητα, αγνοώντας την τρέχουσα κλίση. Τιμές  $m \geq 1$  μπορεί να προκαλέσουν απόκλιση της βελτιστοποίησης καθώς οι ενημερώσεις μπορούν να αυξηθούν εκθετικά. Επομένως, οι τιμές για τις οποίες  $m < 1$  διασφαλίζουν ότι οι παλαιότεροι κλίσεις (**gradients**) έχουν εκθετικά μειούμενη επιρροή, επιτρέποντας στον βελτιστοποιητή να επιταχύνει τη σύγκλιση στην σχετική κατεύθυνση κάθε φορά, να αποσβέσει τις ταλαντώσεις σε άσχετες κατευθύνσεις και να διατηρήσει τη σταθερότητα στη διαδικασία βελτιστοποίησης.

## Chapter 4: A4. Ομαλοποίηση

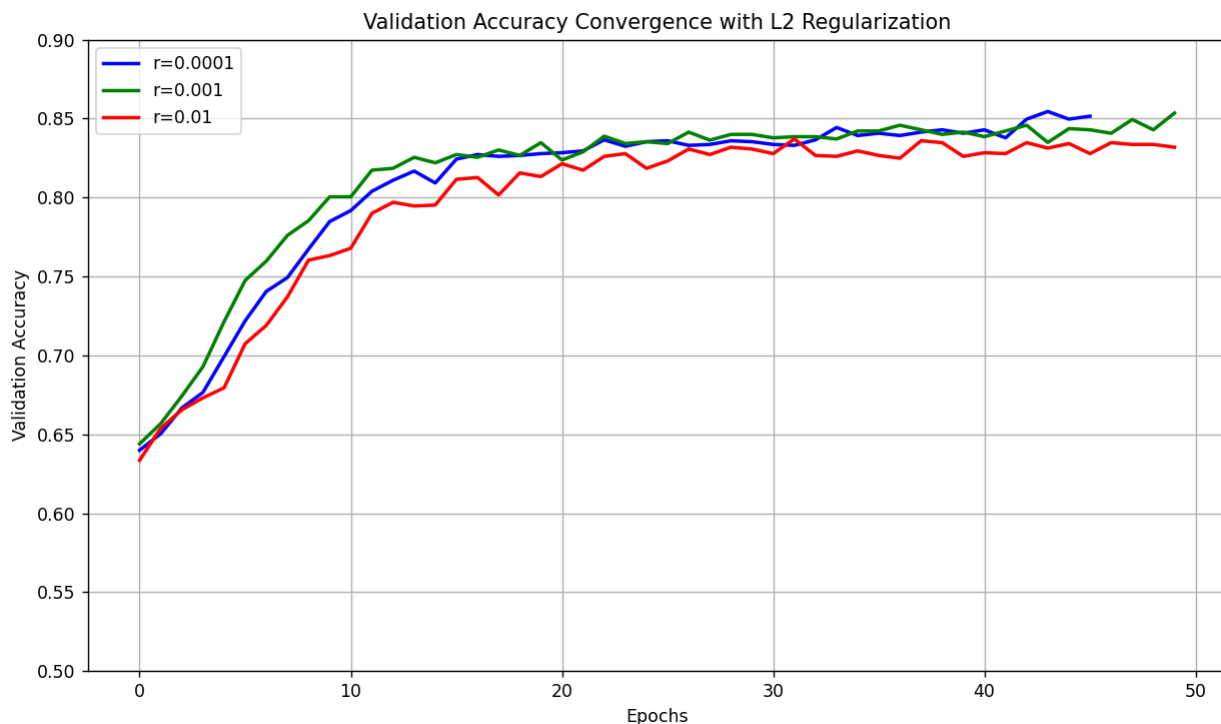
Για το συγκεκριμένο πρόβλημα πρόβλεψης **Alzheimer's**, θα προτιμήσουμε να εφαρμόσουμε **L2 regularization** καθώς επιβάλλει ομαλά (τετραγωνικά) και όχι επιθετικά (γραμμικά όπως το **L1**) ποινές στα



μεγάλα βάρη, λειτουργεί καλύτερα όταν τα περισσότερα χαρακτηριστικά είναι δυνητικά σχετικά (συνηθισμένο στα ιατρικά δεδομένα) και τέλος επειδή τείνει να προσφέρει καλύτερη απόδοση γενίκευσης σε νευρωνικά δίκτυα.

Συντελεστής $r$	CE loss	MSE	Acc
0.0001	0.3936	0.121	0.8325
0.001	0.4114	0.1194	0.8383
0.01	0.5042	0.127	0.8319

Table 3: Regularization Results



#### Ασθενής Κανονικοποίηση ( $r = 0.0001$ ):

Αυτή η τιμή του  $r$ , εφαρμόζει μια πολύ μικρή ποινή στα μεγάλα βάρη, πράγμα που σημαίνει ότι το μοντέλο έχει περισσότερη ελευθερία να προσαρμοστεί στενά στα δεδομένα εκπαίδευσης. Πτυχαίνουμε επίσης τη χαμηλότερη Απώλεια **CE (0.3936)**, υποδηλώνοντας ότι ταιριάζει καλά στα δεδομένα εκπαίδευσης. Ωστόσο, η ακρίβεια επικύρωσής του (**0.8325**) ήταν ελαφρώς χαμηλότερη από την  $r = 0.001$ , γεγονός που υποδηλώνει ότι μπορεί να υπερεκπαιδεύεται—απομνημονεύοντας τον θόρυβο εκπαίδευσης αντί να μαθαίνει γενικά μοτίβα.

#### Μέτρια Κανονικοποίηση ( $r = 0.001$ )

Για αυτή την τιμή του  $r$  πτυχαίνουμε την καλύτερη ισορροπία καθώς έχουμε την υψηλότερη ακρίβεια επικύρωσης (**0.8383**) και το χαμηλότερο **MSE (0.1194)**, πράγμα που σημαίνει ότι το μοντέλο γενικεύει καλύτερα σε μη ορατά δεδομένα. Ενώ η απώλεια **CE (0.4114)** είναι ελαφρώς υψηλότερη από την  $r = 0.0001$ , είναι φυσιολογικό αφού συνήθως ένα καλά κανονικοποιημένο μοντέλο συχνά έχει ελαφρώς υψηλότερη απώλεια εκπαίδευσης επειδή αποφεύγει την υπερπροσαρμογή (**overfitting**).



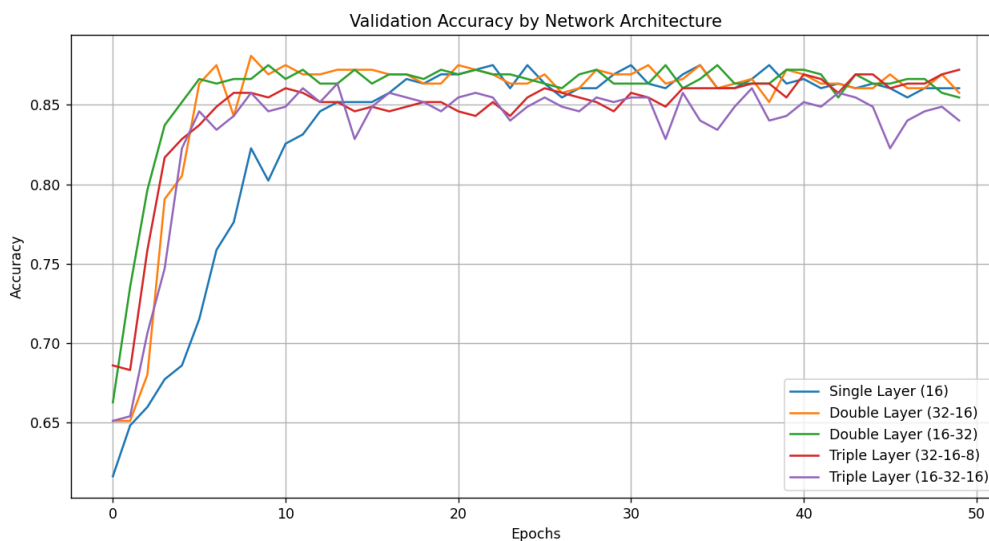
### Ισχυρή Κανονικοποίηση ( $r = 0.01$ )

Στην περίπτωση αυτή τιμωρούνται επιθετικά τα μεγάλα βάρη, αναγκάζοντας το μοντέλο να παραμείνει απλό. Η υψηλότερη απώλεια **CE (0.5042)** και **MSE (0.127)** υποδηλώνουν ότι το μοντέλο υποεκπαιδεύεται και είναι υπερβολικά περιορισμένο για να μάθει αποτελεσματικά. Ενώ η ακρίβεια (**0.8319**) είναι σε σχετικά καλή τιμή, είναι χειρότερη από  $r = 0.001$ , επιβεβαιώνοντας ότι η υπερβολική κανονικοποίηση βλάπτει την απόδοση.

Γενικότερα όλες οι περιπτώσεις έφτασαν σε καλή ακρίβεια (**0,80–0,85**), αλλά το  $r = 0,001$  έφτασε στο υψηλότερο σημείο (**0,85**). Δεν υπήρξαν απότομες διακυμάνσεις στην ακρίβεια, υποδηλώνοντας σταθερή εκπαίδευση.

Το γεγονός ότι το  $r = 0,001$  ξεπέρασε το  $r = 0,0001$  επιβεβαιώνει ότι κάποια κανονικοποίηση βοηθά στη γενίκευση, ενώ η ελαφρά μείωση του  $r = 0,01$  δείχνει ότι η υπερβολική ποσότητα κανονικοποίησης βλάπτει τη μάθηση.

## Chapter 5: A5. Βαθύ Νευρωνικό Δίκτυο



Architecture	CE Loss	MSE	Accuracy
Single Layer (16)	0.4161 ( $\pm 0.0295$ )	0.1192 ( $\pm 0.0090$ )	0.8336 ( $\pm 0.0098$ )
Double Layer (32-16)	0.4285 ( $\pm 0.0250$ )	0.1200 ( $\pm 0.0073$ )	<b>0.8435</b> ( $\pm 0.0045$ )
Double Layer (16-32)	0.4219 ( $\pm 0.0288$ )	0.1195 ( $\pm 0.0078$ )	0.8377 ( $\pm 0.0120$ )
Triple Layer (32-16-8)	0.4541 ( $\pm 0.0293$ )	0.1252 ( $\pm 0.0086$ )	0.8342 ( $\pm 0.0097$ )
Triple Layer (16-32-16)	0.4319 ( $\pm 0.0312$ )	0.1206 ( $\pm 0.0099$ )	0.8301 ( $\pm 0.0105$ )

Table 4: Comparison of Neural Network Architectures

Παρατηρούμε πως η αρχιτεκτονική Διπλού Επιπέδου (**32-16**) πέτυχε υψηλότερη ακρίβεια (**0.8435**) επομένως είχε την καλύτερη προβλεπτική απόδοση, μέτρια απώλεια **CE (0.4285)** - Ελαφρώς υψηλότερη από το μονό επίπεδο αλλά με καλύτερη γενίκευση και συγκρίσιμο **MSE (0.1200)** - Παρόμοιο με απλούστερες



αρχιτεκτονικές.

### 5.a Μοτίβα στην Αρχιτεκτονική

Ένα Μονό Επίπεδο (**16**) απέδωσε εκπληκτικά καλά (ακρίβεια **0.8336**), υποδηλώνοντας ότι το πρόβλημα δεν απαιτεί ακραία πολυπλοκότητα ενώ η προσθήκη επιπέδων βοήθησε μόνο όταν ρυθμίστηκε σωστά (το **32-16** βελτίωσε την ακρίβεια κατά **+1%**).

Τα αποτελέσματα τριών κρυφών επιπέδων έδειξαν ότι και τα δύο υστέρησαν έναντι των διπλών επιπέδων, π.χ. το **32-16-8** έγινε υπερβολικά περιοριστικό (η ακρίβεια έπεσε στο **0.8342**) ενώ το **16-32-16** δυσκολεύτηκε με τη μάθηση χαρακτηριστικών (χαμηλότερη ακρίβεια στο **0.8301**).

Γενικότερα στην περίπτωση του μειούμενου αριθμού κόμβων, έχουμε την καλύτερη συνολική απόδοση: το πρώτο στρώμα των **32** μονάδων καταγράφει ευρείες τάσεις, το δεύτερο στρώμα των **16** μονάδων βελτιώνει τα χαρακτηριστικά και έτσι πετυχαίνουμε την ιδανική ισορροπία μεταξύ χωρητικότητας και γενίκευσης. Για αυξανόμενο αριθμό κρυφών κόμβων είχαμε χαμηλή απόδοση (**-0.6%** ακρίβεια) και παρατηρούμε πως η επέκταση χαρακτηριστικών μπορεί να εισάγει θόρυβο.