ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ & ΒΑΘΙΑ ΜΑΘΗΣΗ

1η ΣΕΙΡΑ ΓΡΑΠΤΩΝ ΑΣΚΗΣΕΩΝ

Ειρήνη Δόντη

ΑΜ: 03119839

8ο εξάμηνο

Αθήνα 2023

Άσκηση 1.1

(α)

Ισχύει ότι x0 = 1, g(x) η συνάρτηση ενεργοποίησης και Δwij = $-\varepsilon \frac{\partial E}{\partial wij}$ με ε τον ρυθμό μάθησης. Οπότε, ισχύουν τα παρακάτω:

Δw13 = $-\varepsilon \frac{\partial \alpha 3}{\partial w13} \frac{\partial E}{\partial \alpha 3} = -\varepsilon \frac{\partial \alpha 3}{\partial w13} \Delta w03 = -\varepsilon(\frac{\partial \alpha 3}{\partial h3} \frac{\partial h3}{\partial w13})$ d3 = -ε g'(h3) x1 d3

Δw23 = $-\varepsilon \frac{\partial \alpha 3}{\partial w23} \frac{\partial E}{\partial \alpha 3} = -\varepsilon \frac{\partial \alpha 3}{\partial w23} \Delta w03 = -\varepsilon(\frac{\partial \alpha 3}{\partial h3} \frac{\partial h3}{\partial w23})$ d3 = -ε g'(h3) x2 d3

Δw14 = $-\varepsilon \frac{\partial \alpha 4}{\partial w14} \frac{\partial E}{\partial \alpha 4} = -\varepsilon \frac{\partial \alpha 4}{\partial w14} \Delta w04 = -\varepsilon(\frac{\partial \alpha 4}{\partial h4} \frac{\partial h4}{\partial w14})$ d4 = -ε g'(h4) x1 d4

Δw24 = $-\varepsilon \frac{\partial \alpha 4}{\partial w24} \frac{\partial E}{\partial \alpha 4} = -\varepsilon \frac{\partial \alpha 4}{\partial w24} \Delta w04 = -\varepsilon(\frac{\partial \alpha 4}{\partial h4} \frac{\partial h4}{\partial w24})$ d4 = -ε g'(h4) x2 d4

(β)

Συμβολίζοντας το σταθμισμένο άθροισμα της εισόδου νευρώνα j με aj = $\sum w_{ij} * y_i$ και την έξοδο yj = g(aj) με χρήση της συνάρτησης ενεργοποίησης $g(x) = \frac{1}{e^{-x}+1}$:

h3:

a3 = w13*x1 + w23*x2 + w03*x0 = 1*1 + (-1)*(-1) +(-1)*(1) = 1

y3 = g(a3) = g(1) = 0.731

h4:

a4 = w14*x1 + w24*x2 + w04*x0 = (-1)*1 + 1*(-1) + (2)*(1) = 0

y4 = g(a4) = g(0) = 0.500

h5:

a5 = w05*x0 + w35*y3 + w45*y4 = (-2)*(1) + (1)*(0,731) + (1)*(0.5) = -0.769

Τιμή εξόδου δικτύου: y5 = g(a5) = 0.317

Το σφάλμα είναι, βάσει της συνάρτησης σφάλματος MSE: e = $\frac{1}{k}\Sigma(t_k - a_k)^2$ = 0.467.

Υπολογισμός ανανεωμένων τιμών βαρών και πολώσεων με χρήση της τεχνικής backpropagation. Ισχύει ότι το βάρος που ενώνει τον νευρώνα εισόδου i με τον κρυφό νευρώνα j είναι $\Delta w_{ij} = \varepsilon * \delta_j * x_i$, με ε τον ρυθμό μάθησης και $\delta_j = y_j * (1 - y_j) * \sum w_{jk} * \delta_k$, ενώ το βάρος που ενώνει τον νευρώνα j με τον νευρώνα k στο τελευταίο επίπεδο είναι $\Delta w_{jk} = \varepsilon * \delta_k * y_j$, με ε τον ρυθμό μάθησης και $\delta_k = 2 * (y_{target} - y_k) * y_k * (1 - y_k)$. Οπότε, ισχύουν τα παρακάτω:

Υπολογίζουμε αρχικά το δ_5: $\delta_5 = 2 * (y_{target} - y_5) * y_5 * (1 - y_5) = 2 * (1 - 0.317) * 0.317 * (1 - 0.317) = 0.296$.

h3:

Δw03 = εδ3x0 = 1*0.058*1 = 0.058, διότι:

$\delta_3 = y_3 * (1 - y_3) * w_{35} * \delta_5$ = 0.731*(1-0.731)*1*0.296= 0.058

Δw13 = εδ3x1 = 1*0.058*1 = 0.058

Δw23 = εδ3x2 = 1*0.058*(-1) = -0.058

h4:

Δw04 = εδ4x0 = 1*0.074*1 = 0.074, διότι:

$\delta_4 = y_4 * (1 - y_4) * w_{45} * \delta_5$ = 0.500*(1-0.500)*1*0.296= 0.074

Δw14 = εδ4x1 = 1*0.074*1 = 0.074

Δw24 = εδ4x2 = 1*0.074*(-1) = -0.074

h5:

Δw05 = εδ5x0 = 1*0.296*1 = 0.296

Δw35 = εδ5y3 = 1*0.296*0.731 = 0.216

Δw45 = εδ5y4 = 1*0.296*0.500 = 0.148

Παρουσιάζουμε τα παλιά και νέα βάρη στο παρακάτω πίνακα:

W_{ij}	$W_{ij,\ old}$	ΔW_{ij}	$W_{ij,\ new} = \Delta W_{ij} + W_{ij,\ old}$
W_{03}	-1	0.058	-0.942
W_{13}	1	0.058	1.058
W_{23}	-1	-0.058	-1.058
W_{04}	2	0.074	2.074
W_{14}	-1	0.074	-0.926
W_{24}	1	-0.074	0.926
W_{05}	-2	0.296	-1.704
W_{35}	1	0.216	1.216
W_{45}	1	0.148	1.148

Υπολογίζουμε όπως πριν τις εισόδους και εξόδους:

h3:

a3 = w13*x1 + w23*x2 + w03*x0 = (1.058)*1 + (-1.058)*(-1) +(-0.942)*(1) = 1.174

y3 = g(a3) = g(1.174) = 0.764

h4:

a4 = w14*x1 + w24*x2 + w04*x0 = (-0.926)*1 + (0.926)*(-1) + 2.074*1 = 0.222

y4 = g(a4) = g(0.222) = 0.555

h5:

a5 = w05*x0 + w35*y3 + w45*y4 = (-1.704)*(1) + (1.216)*(0.764) + (1.148)*(0.555) = -0.138

Τιμή εξόδου δικτύου: y5 = g(a5) = g(-0.138) = 0.466

Το σφάλμα είναι, βάσει της συνάρτησης σφάλματος MSE: e = $\frac{1}{k}\Sigma(t_k - a_k)^2$ = 0.286 < 0.467

Παρατηρούμε ότι η έξοδος y5 είναι πιο κοντά στον στόχο μετά την ανανέωση βαρών, καθώς το σφάλμα μειώθηκε σε σχέση με πριν.

(γ)

Επαναλαμβάνουμε την παραπάνω διαδικασία, συμβολίζοντας το σταθμισμένο άθροισμα της εισόδου νευρώνα j με aj = $\sum w_{ij} * y_i$ και την έξοδο yj = g(aj) με χρήση της συνάρτησης ενεργοποίησης $g(x) = \tanh(x)$:

h3:

a3 = w13*x1 + w23*x2 + w03*x0 = 1*1 + (-1)*(-1) +(-1)*(1) = 1

y3 = g(a3) = g(1) = 0.761

h4:

a4 = w14*x1 + w24*x2 + w04*x0 = (-1)*1 + 1*(-1) + (2)*(1) = 0

y4 = g(a4) = g(0) = 0

h5:

a5 = w05*x0 + w35*y3 + w45*y4 = (-2)*(1) + (1)*(0,762) + (1)*(0) = -1.238

Τιμή εξόδου δικτύου: y5 = g(a5) = -0.845

Το σφάλμα είναι, βάσει της συνάρτησης σφάλματος MSE: e = $\frac{1}{k}\Sigma(t_k - a_k)^2$ = 3.404.

Υπολογισμός ανανεωμένων τιμών βαρών και πολώσεων με χρήση της τεχνικής backpropagation. Χρησιμοποιούμε τους γενικούς τύπους από το ερώτημα 1.α.

Δw05 = d5 = $\frac{\partial E}{\partial y5} = 2(y - y5)(1 - tanh^2(a5)) = 1.055$

Δw03 = d3 = $\frac{\partial E}{\partial y3} = \frac{\partial E}{\partial y5}\frac{\partial y5}{\partial a5}\frac{\partial a5}{\partial y3} = d5(1 - tanh^2(a3)) = 0.443$

Δw04 = d4 = $\frac{\partial E}{\partial y4} = \frac{\partial E}{\partial y5}\frac{\partial y5}{\partial a5}\frac{\partial a5}{\partial y4} = d5(1 - tanh^2(a4)) = 1.055$

Δw13 = d3 = 0.443

Δw23 = -d3 = -0.443

Δw14 = d4 = 1.055

Δw24 = -d4 = -1.055

Δw35 = $d5y3 = 0.804$

Δw45 = $d5y4 = 0$

Παρουσιάζουμε τα παλιά και νέα βάρη στο παρακάτω πίνακα:

W_{ij}	$W_{ij,\ old}$	ΔW_{ij}	$W_{ij,\ new} = \Delta W_{ij} + W_{ij,\ old}$
W_{03}	-1	0.443	-0.557
W_{13}	1	0.443	1.443
W_{23}	-1	-0.443	-1.443
W_{04}	2	1.055	3.055
W_{14}	-1	1.055	0.055
W_{24}	1	−1.055	-0.055
W_{05}	-2	1.055	-0.945
W_{35}	1	0.804	1.804
W_{45}	1	0	1

Υπολογίζουμε όπως πριν τις εισόδους και εξόδους:

h3:

a3 = w13*x1 + w23*x2 + w03*x0 = (1.443)*1 + (-1.443)*(-1) +(-0.557)*(1) = 2.309

y3 = g(a3) = g(2.309) = 0.980

h4:

a4 = w14*x1 + w24*x2 + w04*x0 = (0.055)*1 + (-0.055)*(-1) + 3.055 = 3.165

y4 = g(a4) = g(3.165) = 0.996

h5:

a5 = w05*x0 + w35*y3 + w45*y4 = (-0.945)*(1)+ (1.804)*(0.980) + (1)*(0.996) = 1.819

Τιμή εξόδου δικτύου: y5 = g(a5) = g(1.819) = 0.949

Το σφάλμα είναι, βάσει της συνάρτησης σφάλματος MSE: e = $\frac{1}{k}\Sigma(t_k - a_k)^2$ = 0.003

Παρατηρούμε ότι η έξοδος y5 είναι πιο κοντά στον στόχο μετά την ανανέωση βαρών, καθώς το σφάλμα μειώθηκε σε σχέση με πριν.

Επίσης, σε σχέση με το προηγούμενο ερώτημα, στο οποίο η συνάρτηση ενεργοποίησης είναι η σιγμοειδής, παρατηρούμε ότι το σφάλμα σε αυτή την περίπτωση δε μειώθηκε σε δραματικό βαθμό, όπως σε αυτό το ερώτημα στο οποίο η συνάρτηση ενεργοποίησης είναι η υπερβολική εφαπτομένη. Αυτό σημαίνει ότι η χρήση της υπερβολικής εφαπτομένης ως συνάρτηση ενεργοποίησης, βελτιώνει τη γενίκευση στο επόμενο επίπεδο καλύτερα από τη χρήση σιγμοειδούς συνάρτησης ενεργοποίησης.

(δ)

Η ομαλοποίηση είναι μία τεχνική στην οποία τροποποιείται ένας αλγόριθμος μάθησης ώστε να μειώσει το σφάλμα γενίκευσης, δηλαδή να αποτρέψει την υπερπροσαρμογή (overfitting).

Όταν η παράμετρος ομαλοποίησης αυξάνεται, η επίδραση στην ακρίβεια εξαρτάται από τη τεχνική ομαλοποίησης. Γενικά, *όσο αυξάνεται η παράμετρος ομαλοποίησης*, τείνει να μειώνεται η training accuracy. Αυτό συμβαίνει, καθώς η ομαλοποίηση περιορίζει το μέγεθος των βαρών του μοντέλου. Με αυτόν τον τρόπο, το μοντέλο αναγκάζεται να απλοποιήσει και να μειώσει την υπερπροσαρμογή, γεγονός που *μπορεί να οδηγήσει σε μείωση της training accuracy*.

Επίσης, η υψηλότερη ομαλοποίηση τείνει να βελτιώνει την ακρίβεια των δοκιμών και να αποτρέπει την υπερβολική προσαρμογή. Η μέθοδος ομαλοποίησης βοηθά το μοντέλο να γενικεύει καλύτερα σε ανεξερεύνητα δεδομένα. Αυτό *μπορεί να οδηγήσει σε καλύτερη testing accuracy* και συνεπώς υψηλότερη ακρίβεια δοκιμών (testing accuracy).

Η τεχνική αυτή πρέπει να χρησιμοποιείται με σύνεση, καθώς η υπερβολική χρήση μπορεί να οδηγήσει σε υποπροσαρμογή (underfitting).

Άσκηση 1.2

(a)

Οι διαστάσεις της εξόδου του πρώτου convolutional layer υπολογίζονται ως εξής (θεωρούμε ότι δεν υπάρχει padding):

$W2 = H2 = \frac{W1-F+2P}{S} + 1 = \frac{H1-F+2P}{S} + 1 = \frac{227-11}{4} + 1 = 55$ με W1 = H1 = 227 και F=11 και D2=K=Αριθμός Φίλτρων = 96

Οπότε, η έξοδος του πρώτου convolutional layer θα έχει διαστάσεις 55 x 55 x 96

(b)

Ο αριθμός των units, δηλαδή ο αριθμός των νευρώνων, στο πρώτο convolutional layer είναι ίσος με τον αριθμό των φίλτρων επί το μέγεθος εξόδου, δηλαδή 55*55*96 = 290400 units.

(c)

Ο αριθμός εκπαιδεύσιμων παραμέτρων στο πρώτο convolutional layer του AlexNet υπολογίζεται ως εξής:

Number of parameters = (([Shape of width of the filter]*[Shape of height of the filter]* [Number of filters in the previous layer]+1)*[number of filters] = (11*11*3 + 1)*96 = 34944 παράμετροι.

Δηλαδή το πρώτο convolutional layer του AlexNet έχει 34944 εκπαιδεύσιμες παράμετροι.

(d)

Αν αντικαταστήσουμε το CNN με ένα FeedForward layer με 256 units, ο αριθμός των εκπαιδεύσιμων παραμέτρων είναι: [διάσταση εισόδου + 1(λόγω bias)]*[αριθμός units] = [227*227*3 + 1]*256 = 39574528 εκπαιδεύσιμες παραμέτρους.

Άσκηση 1.3

1. Το μοντέλο θα πρέπει να κωδικοποιήσει την πρόταση “This was horrible” σε μία ακολουθία από διανύσματα 3x1 και να την περάσει από το RNN δίκτυο, για να παραγάγει την τελική κλάση της πρόταση.

   x1 = V[‘This’] = [0, -1, 2]

   x2 = V[‘was’] = V[‘<UNK>’] = [0, 0, 0], η λέξη ‘was’ δεν υπάρχει στο λεξικό οπότε κωδικοποιείται ως UNK.

   x3 = V[‘horrible’] = [-2, -2, 1]

   Η αρχική κρυμμένη κατάσταση του RNN είναι h0 = [0, 0, 0]

Υπολογίζουμε τις κρυφές καταστάσεις για κάθε χρονικό βήμα με τη βοήθεια της συνάρτησης ενεργοποίησης ReLu:

Ισχύει ότι $h_t = ReLu(W_{hh}*h_{t-1} + W_{hx}*x_t + b_h)$ με h0 = 0 και $b_h = 0$ για κάθε h.

$$h1 = ReLu(W_{xh}*x1 + W_{hh}*h0) = ReLu(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0 & -1 & 2\\1 & -2 & 0\end{bmatrix}\begin{bmatrix}0\\-1\\2\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}) =$$

$$ReLu([\begin{bmatrix}5\\2\end{bmatrix}) = \begin{bmatrix}5\\2\end{bmatrix}$$

$$h2 = ReLu(W_{xh}*x2 + W_{hh}*h1) = ReLu(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}5\\2\end{bmatrix} + \begin{bmatrix}0 & -1 & 2\\1 & -2 & 0\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}) =$$

$$ReLu([\begin{bmatrix}5\\2\end{bmatrix}) = \begin{bmatrix}5\\2\end{bmatrix}$$

$$h3 = ReLu(W_{xh}*x3 + W_{hh}*h2) = ReLu(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}5\\2\end{bmatrix} + \begin{bmatrix}0 & -1 & 2\\1 & -2 & 0\end{bmatrix}\begin{bmatrix}-2\\-2\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}) =$$

$$ReLu([\begin{bmatrix}9\\4\end{bmatrix}) = \begin{bmatrix}9\\4\end{bmatrix}$$

Χρησιμοποιούμε τη softmax με την οποία θα λάβουμε την πιθανότητα για κάθε κλάση ταξινόμησης. Η έξοδος του δικτύου είναι η κλάση με τη μεγαλύτερη πιθανότητα. Οπότε, η έξοδος του δικτύου είναι $y = softmax(W_{hy}*h_t + b_y)$

$$y = softmax(\begin{bmatrix}0 & 1\\1 & 0\\2 & -1\end{bmatrix}\begin{bmatrix}9\\4\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}) = softmax(\begin{bmatrix}4\\9\\14\end{bmatrix}) = \begin{bmatrix}0.000045\\0.0067\\0.993\end{bmatrix}$$

Οπότε, το μοντέλο θα ταξινομήσει την πρόταση “This was horrible” στην κλάση 2 (αρνητικό) με πιθανότητα 99%.

2. Από το παραπάνω ερώτημα, παρατηρούμε ότι η ταξινόμηση στην κλάση, κατά μεγάλη πιθανότητα, είναι σωστή. Παρατηρούμε ότι το δίκτυο χρησιμοποιεί κατάλληλα εργαλεία και υπερπαραμέτρους. Το δίκτυο χρησιμοποιεί μία αρχικοποίηση στο 0 για την κρυφή κατάσταση, έναν μοναδιαίο πίνακα Whh και

τη συνάρτηση ενεργοποίησης ReLu για τον υπολογισμό των εσωτερικών καταστάσεων. Το γεγονός ότι το RNN δίκτυο χρησιμοποιεί αναπαράσταση 3x1 για κάθε λέξη και μετατρέπει κείμενα σε μία συναισθηματική κλάση είναι σωστή προσέγγιση. Επίσης, η χρήση μοναδιαίου πίνακα Whh μπορεί να βοηθήσει στην αποφυγή του προβλήματος της εξαφάνισης των gradients. Όμως, από τα παραπάνω, δε σημαίνει απαραίτητα ότι το μοντέλο είναι καλά εκπαιδευμένο, δηλαδή ότι το δείγμα που δίνεται δεν είναι αρκετό για να προσδιορίσει πλήρως το πόσο καλά εκπαιδευμένο είναι το μοντέλο.

3. Αν χρησιμοποιήσουμε average pooling από τα h1, h2 και h3, αντί για το τελευταίο hidden state, τότε η πρόβλεψη θα αλλάξει. Οπότε, η έξοδος του δικτύου είναι $y = \text{softmax}(W_{hy} * h_f + b_y)$ με $h_f = \frac{h_1 + h_2 + h_3}{3} = \begin{bmatrix} 6.333 \\ 2.667 \end{bmatrix}$.

$$y = \text{softmax}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} 6.333 \\ 2.667 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \text{softmax}\left(\begin{bmatrix} 2.667 \\ 6.333 \\ 9.999 \end{bmatrix}\right) = \begin{bmatrix} 0.0006 \\ 0.0249 \\ 0.9744 \end{bmatrix}$$

Παρατηρούμε ότι, όπως και πριν, η πρόταση “This was horrible” ταξινομείται στη σωστή κλάση με πιθανότητα 97%. Η χρήση average pooling, λοιπόν, είναι ένας τρόπος μείωσης της μεροληψίας του μοντέλου, καθώς δίνει περισσότερο χώρο και σε άλλες κλάσεις.

Άσκηση 1.4

(a)

Η διάσταση των χαρακτηριστικών εισόδου xi στον autoencoder είναι ίση με τη διάσταση των αναπαραστάσεων λέξεων, δηλαδή 256 (256x1).

(b)

Η διάσταση των χαρακτηριστικών εξόδου yi στον autoencoder είναι ίση με τη διάσταση των χαρακτηριστικών εισόδου xi στον autoencoder. Εφόσον οι χαρακτηριστικές εισόδου έχουν διάσταση 256, τότε οι χαρακτηριστικές εξόδου θα έχουν επίσης διάσταση 256. Αυτό συμβαίνει καθώς ο αυτοκωδικοποιητής ανακατασκευάζει την είσοδο στο επίπεδο εξόδου.

(c)

Η διάσταση της λανθάνουσας αναπαράστασης (latent representation), που αντιπροσωπεύει τη συμπιεσμένη αναπαράσταση της εισόδου του autoencoder, είναι ίση με τη διάσταση του τρίτου κρυφού στρώματος του autoencoder, δηλαδή 50.

(d)

Χρησιμοποιούμε τον εκπαιδευμένο auto-encoder χρησιμοποιώντας μεγάλο σύνολο δεδομένων κάνοντας fine-tuning για το πρόβλημα της άσκησης 1.3. Η διαδικασία υλοποιείται ως εξής:

Αρχικοποίηση των επιπέδων RNN: Αρχικοποίηση της αρχικής κρυφής κατάστασης h0 χρησιμοποιώντας τα προεκπαιδευμένα βάρη του decoder στον autoencoder. Αυτή η κρυφή κατάσταση μπορεί να χρησιμοποιηθεί ως αρχική κατάσταση για το RNN.

RNN Forward Pass: Με δεδομένες εισόδους x1, x2, x3 και την αρχικοποιημένη κρυφή κατάσταση h0, υπολογίζουμε τις ενημερωμένες εξισώσεις RNN, με τη βοήθεια της συνάρτησης ενεργοποίησης ReLu.

Output Layer: Αφού υπολογίσουμε την κρυφή κατάσταση h3, την περνάμε από ένα πλήρως συνδεδεμένο επίπεδο και εφαρμόζουμε την απαραίτητη συνάρτηση ενεργοποίησης (π.χ. softmax για ταξινόμηση ή γραμμική ενεργοποίηση για παλινδρόμηση), για να δημιουργηθεί η τελική έξοδος y.

Fine-tuning: Ορίζουμε μία κατάλληλη συνάρτηση απώλειας για τη συγκεκριμένη εργασία, όπως το μέσο τετραγωνικό σφάλμα (MSE) για παλινδρόμηση. Εκτελούμε backpropagation για τον υπολογισμό των κλίσεων της απώλειας σε σχέση με τις εκπαιδεύσιμες παραμέτρους του μοντέλου RNN. Ενημερώνουμε τις παραμέτρους, χρησιμοποιώντας έναν αλγόριθμο βελτιστοποίησης SGD ή Adam με βάσει τις υπολογισμένες κλίσεις και έναν καθορισμένο ρυθμό εκμάθησης.

Παρακάτω, παρουσιάζονται οι συναρτήσεις και οι διαστάσεις κάθε στρώματος της αρχιτεκτονικής:

Input Layer: Επίπεδο Εισόδου.

Διάσταση: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Αριθμός χαρακτηριστικών σε κάθε χρονικό βήμα της εισαγωγής).

LSTM/GRU Layer: Προσθήκη επιπέδου για την επεξεργασία της ακολουθίας εισόδου και την εξαγωγή σχετικών χαρακτηριστικών.

Διάσταση: Είσοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Αριθμός χαρακτηριστικών σε κάθε χρονικό βήμα της εισαγωγής), Έξοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Αριθμός κρυφών μονάδων).

Activation Function: Η συνάρτηση ενεργοποίησης που μπορεί να χρησιμοποιηθεί είναι η υπερβολική εφαπτομένη (tanh), λόγω των στρωμάτων LSTM/GRU.

Dense Layer: Προσθήκη πυκνού στρώματος, για να διαμορφωθεί η έξοδος του στρώματος LSTM/GRU.

Διαστάσεις: Είσοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Αριθμός κρυφών μονάδων), Έξοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Μονάδες πυκνού στρώματος).

Activation Function: Η συνάρτηση ενεργοποίησης που μπορεί να χρησιμοποιηθεί είναι η ReLu ή η υπερβολική εφαπτομένη (tanh).

Output Layer: Εφαρμογή ενός πυκνού στρώματος για να δημιουργηθεί το τελικό αποτέλεσμα, με βάση την επεξεργασμένη ακολουθία.

Διάσταση: Είσοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Μονάδες πυκνού στρώματος), Έξοδος: (Αριθμός δειγμάτων που υποβλήθηκαν σε επεξεργασία ανά παρτίδα, Αριθμός χρονικών βημάτων στην ακολουθία εισαγωγής, Διάσταση Εξόδου).

Activation Function: Η συνάρτηση ενεργοποίησης εξαρτάται από την εργασία π.χ. softmax για ταξινόμηση ή γραμμική ενεργοποίηση για παλινδρόμηση.