Τι είναι το batch normalization και γιατί χρσησιμοποιείται στα νευρωνικά δίκτυα;

Το **Batch Normalization** (συχνά συντομογραφείται ως **BatchNorm**) είναι μια τεχνική που χρησιμοποιείται στα **νευρωνικά δίκτυα** για να **επιταχύνει και να σταθεροποιήσει τη διαδικασία εκπαίδευσης**.

**✅ Τι είναι το Batch Normalization;**

Είναι μια διαδικασία που **κανονικοποιεί τις ενεργοποιήσεις (activations)** ενός layer — δηλαδή τις εξόδους των νευρώνων — **ώστε να έχουν μέση τιμή 0 και τυπική απόκλιση 1** μέσα σε κάθε mini-batch κατά την εκπαίδευση.

Πρακτικά: μετασχηματίζει τις εξόδους κάθε layer ώστε:

μB=0,σB2=1\mu\_B = 0,\quad \sigma\_B^2 = 1

**🧪 Πώς δουλεύει βήμα προς βήμα:**

Για κάθε mini-batch:

1. **Υπολογισμός μέσου όρου και διασποράς**:

μB=1m∑i=1mxi,σB2=1m∑i=1m(xi−μB)2\mu\_B = \frac{1}{m} \sum\_{i=1}^{m} x\_i,\quad \sigma\_B^2 = \frac{1}{m} \sum\_{i=1}^{m} (x\_i - \mu\_B)^2

όπου xix\_i οι ενεργοποιήσεις του batch.

1. **Κανονικοποίηση των τιμών**:

x^i=xi−μBσB2+ϵ\hat{x}\_i = \frac{x\_i - \mu\_B}{\sqrt{\sigma\_B^2 + \epsilon}}

(όπου ϵ\epsilon είναι μικρό θετικό για σταθερότητα)

1. **Κλιμάκωση και μετατόπιση** (για να μη χάνεται η εκφραστικότητα του μοντέλου):

yi=γx^i+βy\_i = \gamma \hat{x}\_i + \beta

Όπου:

* + γ\gamma: παράγοντας κλιμάκωσης (learnable)
  + β\beta: μετατόπιση (learnable)

**🎯 Γιατί χρησιμοποιείται;**

**🔸 1. Μειώνει το "Internal Covariate Shift"**

* Κατά την εκπαίδευση, οι διανομές των ενεργοποιήσεων αλλάζουν από layer σε layer.
* Το BatchNorm **σταθεροποιεί αυτές τις διανομές**, ώστε τα επόμενα layers να λαμβάνουν εισόδους με σταθερό εύρος τιμών.

**🔸 2. Επιτρέπει μεγαλύτερα learning rates**

* Βελτιώνει τη σταθερότητα της εκπαίδευσης ➜ επιτρέπει να χρησιμοποιήσεις **μεγαλύτερα learning rates χωρίς να "εκραγεί" το μοντέλο**.

**🔸 3. Μειώνει την εξάρτηση από weight initialization**

* Με το BatchNorm, το μοντέλο είναι **λιγότερο ευαίσθητο** στην αρχική επιλογή βαρών.

**🔸 4. Δρα ως ελαφριά μορφή regularization**

* Η κανονικοποίηση σε mini-batches εισάγει κάποια **τυχαία μεταβολή** (noise), παρόμοια με το Dropout ➜ μικρότερη υπερπροσαρμογή (overfitting).

**📌 Πού χρησιμοποιείται;**

* **Μεταξύ dense/fully-connected layers**
* **Μετά από convolutional layers** (συνήθως πριν την ReLU)
* Συχνά: Conv → BatchNorm → ReLU (το πιο κλασικό pattern)

**⚖️ BatchNorm vs Dropout**

|  | **Batch Normalization** | **Dropout** |
| --- | --- | --- |
| Τύπος | Κανονικοποίηση | Regularization (τυχαία απώλεια) |
| Σκοπός | Σταθεροποίηση & ταχύτητα | Αποφυγή υπερπροσαρμογής |
| Χρόνος εφαρμογής | Εκπαίδευση + Πρόβλεψη | Συνήθως μόνο στην εκπαίδευση |

**🧠 Παράδειγμα σε Keras / TensorFlow**

model.add(Dense(128))

model.add(BatchNormalization())

model.add(Activation('relu'))

**📝 Συμπερασματικά**

Το **Batch Normalization** είναι ένα ισχυρό εργαλείο που:

* Βοηθά στην **ταχύτερη και σταθερότερη εκπαίδευση**
* Μειώνει την ανάγκη για τέλειο tuning hyperparameters
* Βελτιώνει τη **γενίκευση** και την **απόδοση του μοντέλου**

Αν θες, μπορώ να σου δείξω **γράφημα** που απεικονίζει πώς αλλάζει η κατανομή των ενεργοποιήσεων **με και χωρίς BatchNorm**. Θες να το ετοιμάσω;