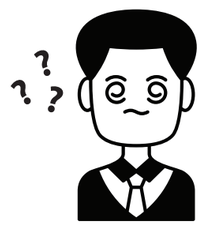

Neural Networks For People Who Get Confused Easily

by James McCammon (A fellow who gets confused easily)

Forward pass basics

Top level view (simplified version)

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}\\\text{Number of neurons}}} = \mathbf{ReLU}\left(\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{X} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{m}\\\text{Number of features}}} \cdot \underbrace{\left[\begin{matrix}\\ \mathbf{\Omega}^{\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{m}\\\text{Number of weights}}}_{\mathbf{p}} + \underbrace{\left[\begin{matrix}\\ \mathbf{b} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}\\\text{Number of neurons}}}_{\mathbf{1}} \right)$$

The hidden layer activation matrix

The input matrix

The weights matrix

The bias vector

Top level view (crazy annotated version)

Often called **A** for "activation matrix"

The hidden layer activation matrix

Each entry in **H** tells us how one neuron (a column in $\mathbf{\Omega}^{\mathbf{T}}$) processed one input (a row in the **X**)

How neuron *j* processed input *i* after activation

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H}^{[\ell+1]} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[\ell]}\\\text{Number of neurons}}} = \mathbf{ReLU}\left(\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H}^{[\ell]} \cdot \mathbf{\Omega}^{[\ell]\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[\ell]}\\\text{Number of neurons}\\\text{in the current layer}}} + \underbrace{\left[\begin{matrix}\\ \mathbf{b} \\ \\ \end{matrix}\right]}_{\mathbf{p}^{[\ell]}}_{\mathbf{1}} \right)$$

The post-activation matrix, produced as the end result of every layer, acting as the input to the next layer. **H** is not necessarily square (usually it is not).

The pre-activation matrix, often denoted as **Z**.

$\mathbf{Z} = \mathbf{X}\mathbf{\Omega}^{\mathbf{T}} + \mathbf{b}$

The bias vector is really added in this direction

b is "broadcasted" to $\mathbf{X}\cdot\mathbf{\Omega}^{\mathbf{T}}$ (See notes).

The number of weights per neuron in **Ω** equals the number of input features.

Not necessarily square (usually it is not).

We transpose the weights matrix **Ω**.

We designate the layer with brackets. The first layer is $\ell = 0$.

We right multiply by the weights

Defined as: $\texttt{max(0, } Z_{ij}\texttt{)}$ Applied to every entry of **Z**.

Notes

1. We can initialize our weights as (m x p) and save a transform operation.

2. The form $\mathbf{X}\mathbf{\Omega}^{\mathbf{T}} + \mathbf{b}$ is a non-linear affine transform and allows us to fit complex non-linear functions. We can easily prove it is affine by noting that it moves the origin. All linear functions must preserve the origin.

   We can use the general form: $\mathbf{Wx} + \mathbf{b}$, where x is now an input vector. If we set (x_1, x_2) equal to the origin we see the origin moves from (0, 0) to (3, 4) and is therefore not preserved.

$$\underset{\mathbf{W}}{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}} \underset{\mathbf{x}}{\begin{bmatrix}0\\0\end{bmatrix}} + \underset{\mathbf{b}}{\begin{bmatrix}3\\4\end{bmatrix}} = \underset{\mathbf{Wx}+\mathbf{b}}{\begin{bmatrix}3\\4\end{bmatrix}}$$

3. Implicitly we are using an outer product when we add **b**. As in the below where $\otimes$ denotes the outer product. See next slide for details.

   $\mathbf{Z} = \mathbf{X}\mathbf{\Omega}^{\mathbf{T}} + \mathbf{1}_{\mathbf{n}} \otimes \mathbf{b}^{\mathbf{T}}$

4. How does layer numbering work? The input and associated weights are considered layer 0 (ℓ=0). After ReLU activation, the result is **H**, which is fed into the next hidden layer as the input. Thus H is designated as Layer 1 (ℓ=1), but note that it takes its dimensions from layer 0. The final layer is the output layer, L. In practice L is just a number. Thus there are L-2 hidden layers.

Multi-layer view

The first input (in Layer 0) is the input matrix.

Z is always created as an intermediate matrix.

Layer 0: $\mathbf{H}^{[1]} = \mathbf{ReLU}(\underbrace{\mathbf{X}\cdot\mathbf{\Omega}^{[0]\mathbf{T}} + \mathbf{b}^{[0]}}_{\mathbf{Z}^{[0]}})$

Subsequent layers intake the output of the previous layer.

Layer 1: $\mathbf{H}^{[2]} = \mathbf{ReLU}(\underbrace{\mathbf{H}^{[1]}\cdot\mathbf{\Omega}^{[1]\mathbf{T}} + \mathbf{b}^{[1]}}_{\mathbf{Z}^{[1]}})$

Layer 2: $\mathbf{H}^{[3]} = \mathbf{ReLU}(\underbrace{\mathbf{H}^{[2]}\cdot\mathbf{\Omega}^{[2]\mathbf{T}} + \mathbf{b}^{[2]}}_{\mathbf{Z}^{[2]}})$

⋮ **Layer 3 to Layer L-1**

The output of the last layer is used to calculate the loss according to some specified loss function.

Layer L: $\mathbf{Loss} = \mathbf{f}_{\mathbf{Loss}}(\underbrace{\mathbf{H}^{[L]}\cdot\mathbf{\Omega}^{[L]\mathbf{T}} + \mathbf{b}^{[L]}}_{\mathbf{Z}^{[L]}})$

What the heck are the dimensions?

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{X} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{f}\\\text{\# of features}}} \qquad \underset{\substack{\mathbf{f}\\\text{\# of features}}}{\left[\begin{matrix}\\ \mathbf{\Omega}^{[0]\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[0]}\\\text{\# of neurons}\\\text{in Layer 0}}} \qquad \underset{\substack{\mathbf{p}^{[0]}\\\text{\# of neurons}\\\text{in Layer 0}}}{\left[\begin{matrix}\\ \mathbf{b}^{[0]} \\ \\ \end{matrix}\right]}_{\mathbf{1}}$$

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H}^{[1]} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[1]}\\\text{\# of neurons}\\\text{in Layer 1}}} \qquad \underset{\substack{\mathbf{p}^{[0]}\\\text{\# of neurons}\\\text{in Layer 0}}}{\left[\begin{matrix}\\ \mathbf{\Omega}^{[1]\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[1]}\\\text{\# of neurons}\\\text{in Layer 1}}} \qquad \underset{\substack{\mathbf{p}^{[1]}\\\text{of neurons}\\\text{in Layer 1}}}{\left[\begin{matrix}\\ \mathbf{b}^{[1]} \\ \\ \end{matrix}\right]}_{\mathbf{1}}$$

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H}^{[2]} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[2]}\\\text{\# of neurons}\\\text{in Layer 2}}} \qquad \underset{\substack{\mathbf{p}^{[1]}\\\text{\# of neurons}\\\text{in Layer 1}}}{\left[\begin{matrix}\\ \mathbf{\Omega}^{[2]\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[2]}\\\text{\# of neurons}\\\text{in Layer 2}}} \qquad \underset{\substack{\mathbf{p}^{[2]}\\\text{of neurons}\\\text{in Layer 2}}}{\left[\begin{matrix}\\ \mathbf{b}^{[2]} \\ \\ \end{matrix}\right]}_{\mathbf{1}}$$

⋮ **Layer 3 to Layer L-1**

$$\underset{\substack{\text{n}\\\text{Batch size}}}{\left[\begin{matrix}\\ \mathbf{H}^{[L]} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[L]}\\\text{\# of neurons}\\\text{in Layer L}}} \qquad \underset{\substack{\mathbf{p}^{[L]}\\\text{\# of neurons}\\\text{in Layer L}}}{\left[\begin{matrix}\\ \mathbf{\Omega}^{[L]\mathbf{T}} \\ \\ \end{matrix}\right]}_{\substack{\mathbf{p}^{[L]}\\\text{of neurons}\\\text{in Layer 0}}} \qquad \underset{\substack{\mathbf{p}^{[L]}\\\text{of neurons}\\\text{in Layer 0}}}{\left[\begin{matrix}\\ \mathbf{b}^{[L]} \\ \\ \end{matrix}\right]}_{\mathbf{1}}$$