

# Custom Neural Networks

<https://github.com/lucacareddu/Custom-Neural-Networks>

## 1 1-layer Transformer Encoder

### Forward

$$X \in \mathcal{R}^{n \times m}, W_1 \in \mathcal{R}^{m \times d_1}, \text{cls\_tok} \in \mathcal{R}^{d_1}, W_{\{Q,K,V,T\}} \in \mathcal{R}^{d_1 \times d_2}, W_2 \in \mathcal{R}^{d_2 \times k}, y \in \mathcal{R}^k$$

$$h_{pre_1} = \frac{X - \mathbb{E}(X)}{\sqrt{\mathbb{E}(X - \mathbb{E}X)^2}} \in \mathcal{R}^{n \times m} \quad (1)$$

$$h_1 = h_{pre_1} W_1 \in \mathcal{R}^{n \times d_1} \quad (2)$$

$$h_{pre_2} = \text{concat}(h_1, \text{cls\_tok}^\top) \in \mathcal{R}^{(n+1) \times d_1} \quad (3)$$

$$\{Q, K, V, T\} = h_{pre_2} W_{\{Q,K,V,T\}} \in \mathcal{R}^{(n+1) \times d_2} \quad (4)$$

$$S = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \in \mathcal{R}^{(n+1) \times (n+1)} \quad (5)$$

$$h_2 = SV + T \in \mathcal{R}^{(n+1) \times d_2} \quad (6)$$

$$z = h_{2(n+1)} W_2 \in \mathcal{R}^k \quad (7)$$

$$s = \text{Softmax}(z) \in \mathcal{R}^k \quad (8)$$

$$\mathcal{L} = -(y \odot \ln s) \mathbf{1}_k^\top \in \mathcal{R} \quad (9)$$

$$\begin{aligned} \mathcal{L}(X, W_1, \text{cls\_tok}, W_{\{Q,K,V,T\}}, W_2, y) &= \\ CE(y, SM(\text{SelfAttention}(\text{concat}((XW_1), \text{cls\_tok}), W_{\{Q,K,V,T\}})_{[\text{seq\_len+1}, :]} W_2)) \end{aligned}$$

### Backward

$$\frac{\partial \mathcal{L}}{\partial W_2} = h_{2(n+1)}^\top (z - y) \quad (1)$$

$$\frac{\partial \mathcal{L}}{\partial W_T} = h_{pre_2(n+1)}^\top (z - y) W_2^\top \quad (2)$$

$$\frac{\partial \mathcal{L}}{\partial W_V} = h_{pre_2}^\top S_{(n+1)}^\top (z - y) W_2^\top \quad (3)$$

Now, with

$$\frac{\partial \mathcal{L}}{\partial S_{(n+1)}} = (\mathbf{z} - \mathbf{y}) \mathbf{W}_2^\top \mathbf{V}^\top = \nabla_{S_{n+1}} \quad (4)$$

$$\frac{\partial S_{(n+1)}}{\partial Q_{(n+1)} K^\top} = \frac{1}{\sqrt{d_k}} (diag(\mathbf{S}_{(n+1)}) - \mathbf{S}_{(n+1)}^\top \mathbf{S}_{(n+1)}) \quad (5)$$

it follows that

$$\frac{\partial \mathcal{L}}{\partial Q_{(n+1)} K^\top} = \frac{\partial \mathcal{L}}{\partial S_{(n+1)}} \frac{\partial S_{(n+1)}}{\partial Q_{(n+1)} K^\top} \quad (6)$$

$$= \frac{1}{\sqrt{d_k}} \mathbf{S}_{(n+1)} \odot (\nabla_{S_{n+1}} - \nabla_{S_{n+1}} \mathbf{S}_{(n+1)}^\top) \quad (7)$$

$$= \nabla_{Q_{n+1} K^\top} \quad (8)$$

Then we have

$$\frac{\partial \mathcal{L}}{\partial W_K} = \mathbf{h}_{pre_2}^\top (\nabla_{Q_{n+1} K^\top})^\top \mathbf{Q}_{(n+1)} \quad (9)$$

$$\frac{\partial \mathcal{L}}{\partial W_Q} = \mathbf{h}_{pre_2(n+1)}^\top \nabla_{Q_{n+1} K^\top} \mathbf{K} \quad (10)$$

Given that

$$\frac{\partial \mathcal{L}}{\partial Q_{(n+1)}} = \nabla_{Q_{n+1} K^\top} \mathbf{K} = \nabla_{Q_{n+1}} \quad (11)$$

$$\frac{\partial \mathcal{L}}{\partial K} = (\nabla_{Q_{n+1} K^\top})^\top \mathbf{Q}_{(n+1)} = \nabla_K \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial V} = \mathbf{S}_{(n+1)}^\top (\mathbf{z} - \mathbf{y}) \mathbf{W}_2^\top = \nabla_V \quad (13)$$

$$\frac{\partial \mathcal{L}}{\partial T} = (\mathbf{z} - \mathbf{y}) \mathbf{W}_2^\top = \nabla_T \quad (14)$$

we have

$$\frac{\partial \mathcal{L}}{\partial \text{cls\_tok}} = \nabla_{Q_{n+1}} \mathbf{W}_Q^\top + \nabla_{K_{(n+1)}} \mathbf{W}_K^\top + \nabla_{V_{(n+1)}} \mathbf{W}_V^\top + \nabla_T \mathbf{W}_T^\top \quad (15)$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = \mathbf{h}_{pre_1}^\top (\nabla_{K_{(0:n)}} \mathbf{W}_K^\top + \nabla_{V_{(0:n)}} \mathbf{W}_V^\top) \quad (16)$$

## 2 Vanilla 2-layers NN

### Forward

$$X \in \mathcal{R}^{n \times m}, W_1 \in \mathcal{R}^{d \times m}, W_2 \in \mathcal{R}^{k \times dn}, B_1 \in \mathcal{R}^{d \times n}, B_2 \in \mathcal{R}^k, y \in \mathcal{R}^k$$

$$pre_{a_1} = \frac{X - \mathbb{E}(X)}{\sqrt{\mathbb{E}(X - \mathbb{E}X)^2}} \in \mathcal{R}^{n \times m} \quad (1)$$

$$a_1 = W_1 pre_{a_1}^\top + B_1 \in \mathcal{R}^{d \times n} \quad (2)$$

$$h_1 = \text{ReLU}(a_1) \in \mathcal{R}^{d \times n} \quad (3)$$

$$pre_{a_2} = Flatten(h_1) \in \mathcal{R}^{dn} \quad (4)$$

$$a_2 = (W_2 pre_{a_2}^\top)^\top + B_2 \in \mathcal{R}^k \quad (5)$$

$$h_2 = a_2 I_k \in \mathcal{R}^k \quad (6)$$

$$s = Softmax(h_2) \in \mathcal{R}^k \quad (7)$$

$$\mathcal{L} = -(y \odot \ln s) \mathbf{1}_k^\top \in \mathcal{R} \quad (8)$$

$$\mathcal{L}(X, W_1, W_2, y) = \text{CrossEntropy}(y, Softmax(Flatten(\text{ReLU}(W_1 X + B1)) W_2^\top + B2))$$

### Backward

We have

$$\left( \frac{\partial \mathcal{L}}{\partial W_2} \right)_{k \times dn} = \left( \frac{\partial \mathcal{L}}{\partial s} \right)_k \left( \frac{\partial s}{\partial h_2} \right)_{k \times k} \left( \frac{\partial h_2}{\partial a_2} \right)_{k \times k} \left( \frac{\partial a_2}{\partial W_2} \right)_{k \times k \times dn} \quad (1)$$

or, equivalently,

$$\left( \frac{\partial \mathcal{L}}{\partial W_2} \right)_{k \times dn} = \left( \frac{\partial \mathcal{L}}{\partial h_2} \right)_k \left( \frac{\partial h_2}{\partial a_2} \right)_{k \times k} \left( \frac{\partial a_2}{\partial W_2} \right)_{k \times k \times dn} \quad (2)$$

In a simplified vectorized form, we have

$$\frac{\partial \mathcal{L}}{\partial W_2} = ((s - y) \odot \mathbf{1}_k)^\top pre_{a_2} \quad (3)$$

Meanwhile

$$\begin{aligned} \left( \frac{\partial \mathcal{L}}{\partial W_1} \right)_{d \times m} &= \left( \frac{\partial \mathcal{L}}{\partial h_2} \right)_k \left( \frac{\partial h_2}{\partial a_2} \right)_{k \times k} \left( \frac{\partial a_2}{\partial pre_{a_2}} \right)_{k \times dn} \left( \frac{\partial pre_{a_2}}{\partial h_1} \right)_{dn \times d \times n} \\ &\quad \left( \frac{\partial h_1}{\partial a_1} \right)_{d \times n \times d \times n} \left( \frac{\partial a_1}{\partial W_1} \right)_{d \times n \times d \times m} \end{aligned} \quad (4)$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial W_1} = ((\mathbf{s} - \mathbf{y}) \odot \mathbf{1}_k \mathbf{W}_2)_{dn \rightarrow d \times n} \odot \left( \frac{\partial \text{ReLU}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \right)_{d \times n} \mathbf{pre}_{\mathbf{a}_1} \quad (5)$$

Biases gradients are computed as follows:

$$\frac{\partial \mathcal{L}}{\partial B_2} = (\mathbf{s} - \mathbf{y}) \odot \mathbf{1}_k \quad (6)$$

$$\frac{\partial \mathcal{L}}{\partial B_1} = ((\mathbf{s} - \mathbf{y}) \odot \mathbf{1}_k \mathbf{W}_2)_{dn \rightarrow d \times n} \odot \left( \frac{\partial \text{ReLU}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \right)_{d \times n} \quad (7)$$