1 Deep Learning

1. We are training fully connected network with two hidden layers to predict housing prices. Inputs are 100-dimensional, and have several features such as the number of square feet, the median family income, etc. The first hidden layer has 1000 activations. The second hidden layer has 10 activations. The output is a scalar representing the house price.

   Assuming a vanilla network (e.g., no layer normalization, no learnable activation function parameters, etc.), how many parameters does this network have?

2. Let us analyze the linearity and convexity of deep neural networks.

   Recall that a function $g(x)$ is linear if for all $a, b \in \mathbb{R}$, $g(ax + b) = ag(x) + b$. Say that a function $f : \mathbb{R} \to \mathbb{R}$ is convex if and only if $f((1-t)x + ty) \leq (1-t)f(x) + tf(y)$ for $t \in [0, 1]$ and all x, y. Select all that are true.

   (a) The following fully connected network without activation functions is linear: $g_3(g_2(g_1(x)))$, where $g_i(x) = W_i x$ and W_i are matrices
   (b) Leaky ReLU $= \max\{0.01x, x\}$ is convex
   (c) A combination of ReLUs such as $\text{ReLU}(x) - \text{ReLU}(x-1)$ is convex
   (d) ResNet-50, which has ReLU activations, is nonlinear and convex

3. Consider the properties (a) convex, (b) monotonically nondecreasing, (c) invertible, (d) has a zero gradient at multiple points. For the sigmoid, ReLU, and GELU activation functions, indicate which properties are true.

4. Write a simple PyTorch command to produce a 10×5 Gaussian matrix with each entry i.i.d. sampled from $\mathcal{N}(\mu = 5, \sigma^2 = 16)$. Write a separate command to produce a 10×10 uniform matrix with each entry i.i.d. sampled from $U[-1, 1)$.

5. The following options have a convolution kernel and a description of what the kernel does to a natural image. Select options that correctly describe the effect of the kernel on the image after convolution. (Note, a vertical line has changes in the vertical direction.)

   (a) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; this would not change the image

(b) $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$; this would not change the image

(c) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$; this would not change the image

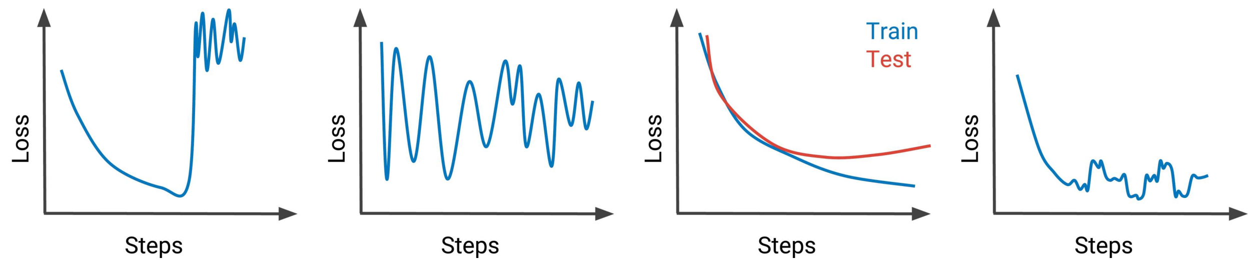
(d) $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$; this would make the image smoother

(e) $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$; this emphasize/detect horizontal changes in the image

(f) $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$; this emphasize/detect vertical changes in the image

(g) $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$; this would make the image smoother

6. Assign the best "possible action" to exactly one training curve. Use a possible explanation at most once. Assume these are learning curves for a model trained to classify CIFAR images using PyTorch trained with a typical cross entropy loss. Choose the most fitting answer.



(a) Spiking curve (b) Wild curve (c) Test goes up (d) Cyclic

Figure 1: Learning curves

Here are possible actions:
(1) Use the squared loss instead of the cross entropy loss
(2) Shuffle your batches
(3) Set the batch size to one
(4) Set the batch size to the size of the training set
(5) Use data augmentation

(6) Check for division by zero or logarithms of small numbers
(7) Use a loss like the area under the curve which is threshold-invariant
(8) Decrease the learning rate

What is the answer for each curve?

7. When we optimize the parameters of a neural network, we are searching for the best function f that is consistent with a particular neural architecture. For example, suppose that our network takes an input $x \in \mathbb{R}^{100}$, outputs a vector $f(x) \in \mathbb{R}^{24}$, and has one hidden layer $z \in \mathbb{R}^{40}$ followed by a `tanh` activation. The set of all functions consistent with that architecture is

$$\mathcal{F} = \{f \mid f(x) = W_2\texttt{tanh}(W_1 x)\,, x \in \mathbb{R}^{100}\,, W_1 \in \mathbb{R}^{40\times 100}\,, W_2 \in \mathbb{R}^{24\times 40}\}\,.$$

We say that two network architectures α and β have the same *expressivity* if the set of functions $\mathcal{F}_\alpha$ and $\mathcal{F}_\beta$ which are consistent with each architecture are equivalent, i.e. $\mathcal{F}_\alpha = \mathcal{F}_\beta$. Which **two** architectures below which have the same expressivity?

(a) Two linear layers, `ReLU`, two linear layers.

$$\begin{aligned}\mathcal{F}_a = \{f \mid f(x) &= W_4 W_3 \texttt{ReLU}(W_2 W_1 x)\,,\\ &x \in \mathbb{R}^{100}, W_1 \in \mathbb{R}^{60\times 100}, W_2 \in \mathbb{R}^{40\times 60}, W_3 \in \mathbb{R}^{32\times 40}, W_4 \in \mathbb{R}^{24\times 32}\}\end{aligned}$$

(b) Two linear layers, `tanh`, one linear layer, `ReLU`

$$\begin{aligned}\mathcal{F}_b = \{f \mid f(x) &= \texttt{ReLU}(W_3\texttt{tanh}(W_2 W_1 x))\,,\\ &x \in \mathbb{R}^{100}, W_1 \in \mathbb{R}^{60\times 100}, W_2 \in \mathbb{R}^{40\times 60}, W_3 \in \mathbb{R}^{24\times 40}\}\end{aligned}$$

(c) One linear layer, `ReLU`, two linear layers

$$\begin{aligned}\mathcal{F}_c = \{f \mid f(x) &= W_3 W_2 \texttt{ReLU}(W_1 x)\,,\\ &x \in \mathbb{R}^{100}, W_1 \in \mathbb{R}^{40\times 100}, W_2 \in \mathbb{R}^{39\times 40}, W_3 \in \mathbb{R}^{24\times 39}\}\end{aligned}$$

(d) Two linear layers, `ReLU`, one linear layer, `ReLU`

$$\begin{aligned}\mathcal{F}_d = \{f \mid f(x) &= \texttt{RELU}(W_3\texttt{ReLU}(W_2 W_1 x))\,,\\ &x \in \mathbb{R}^{100}, W_1 \in \mathbb{R}^{60\times 100}, W_2 \in \mathbb{R}^{40\times 60}, W_3 \in \mathbb{R}^{24\times 40}\}\end{aligned}$$

8. Consider a convolution layer that processes an input image of size $256 \times 256 \times 3$ and where the output has one channel. If the convolution layer contains a single 1×1 convolution filter, how many parameters are there including bias?

9. Consider this hypothetical optimizer:

$$
\begin{aligned}
&n \leftarrow n+1 \qquad \# \ (n \text{ initialized at } 0) \\
&g \leftarrow \frac{1}{n}\nabla\theta \sum_{i=1}^{N} \ell(\theta; x_i, y_i) + \frac{n-1}{n} g \\
&\theta \leftarrow \theta - \alpha g
\end{aligned}
$$

How does this optimizer differ from SGD with momentum?

(a) This optimizer gives less weight to "important" gradient dimensions, which could slow time until convergence.

(b) This optimizer's running average gives more weight to older gradients, which could slow or prevent convergence.

(c) The magnitude of the optimizer's updates is guaranteed to converge to zero more rapidly.

10. Suppose we have a one hidden layer ReLU network $f(x) = w^\top \max(Ax, 0)$ with input $x \in \mathbb{R}^d$, first weight matrix $A \in \mathbb{R}^{k\times d}$, and second weight vector $w \in \mathbb{R}^k$. What is the gradient of $f(x)$ with respect to the input x?

(a) $\max(A^\top w, 0)$

(b) $\max(A^\top, 0)$

(c) $A^\top Zw$ where $Z \in \mathbb{R}^{k\times k}$ is a diagonal matrix such that $(Z)_{ii}$ is equal to 1 if $(Ax)_i > 0$ and 0 if $(Ax)_i \leq 0$.

(d) $A^\top w$

(e) $A^\top Zw$ where $Z \in \mathbb{R}^{k\times k}$ is a diagonal matrix such that $(Z)_{ii}$ is equal to 1 if $(x)_i > 0$ and 0 if $(x)_i \leq 0$.

11. Consider the following plots related to scaling laws, taken from the scaling laws paper (left) and the GPT-3 (right) paper.

Which options are true?

(a) In both sets of plots, the first batch of the largest model uses more compute than the entire training run of the smallest model.

(b) In the right plot, the loss lower bound is $L = 2.57 \cdot C^{-0.048}$. If the 2.57 were replaced by 3, the slope of the line representing L would change.

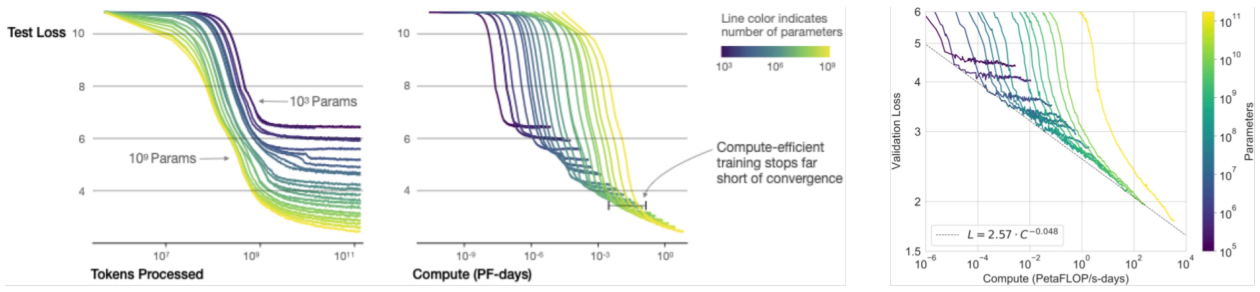


Figure 2: Scaling law plots.

(c) Some of the models in these plots are showing signs of overfitting.

(d) The left plot shows that larger models require fewer training tokens to reach the same performance as smaller models.

12. Suppose that when performing attention, we have the following keys and values:

Keys:

$$\left\{ \begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Values:

$$\left\{ \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \right\}$$

We want to compute the attention embedding using these keys and values for the following query:

$$\begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}$$

Which of the following is the correct attention embedding? **To simplify calculations, replace softmax with argmax. For example, softmax$([-1, 1, 0])$ would instead be argmax$([-1, 1, 0]) = [0, 1, 0]$.**

(a)

$$\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}$$

(d)

$$\begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix}$$

13. What is the time complexity of using beam search to generate a sequence of length T, if we use a beam width of K? Treat the vocabulary size and run time for each forward pass of our model as constants. Beam width refers to the number of candidate sequences we maintain.

    (a) $\mathcal{O}(K^2T)$
    (b) $\mathcal{O}(K^T)$
    (c) $\mathcal{O}(T^K)$
    (d) $\mathcal{O}(KT)$

14. Mark true for all of the following statements that are correct about LSTMs and vanilla RNNs, and false otherwise.

    (a) RNNs use backpropagation through time to compute gradients because a forward pass involves a time component.
    (b) Using gating functions in LSTMs help prevent vanishing gradients.
    (c) Gradient clipping cannot be used to remedy the exploding gradient problem in vanilla RNNs.
    (d) LSTMs can model long-range dependencies that vanilla RNNs cannot.

15. Mark true for all of the following statements that are correct about Transformers, and false otherwise.

    (a) Unlike with RNNs, the amount of learnable parameters in a transformer scales with the maximum sequence length of inputs it is trained on.
    (b) If we remove all of the feedforward layers in a standard transformer, each output of our model at each timestep is a linear combination of the inputs.
    (c) Without positional encodings, if you permute the input sequence to a transformer encoder, the resulting output sequence will be the output sequence of the original input, except permuted in the same way.

(d) In a single multi-head attention layer, the operations for each head can be run in parallel to the other heads (e.g. the operations for one head do not depend on the others).

16. You want to build a text classifier that takes in a sequence of text and outputs its sentiment. You want to use a pretrained model to extract a fixed-size representation from the input text sequence, and train a linear classifier on this representation. You would like this representation to capture information from the entire input sequence. For an input sequence with 10 tokens (including start/stop tokens), mark true for the following representations which could satisfy this property, and false otherwise.

    (a) The 5th embedding in the final layer of embeddings of BERT.

    (b) The 10th embedding in the second layer of embeddings of BERT.

    (c) The 5th embedding in the final layer of embeddings of a unidirectional Transformer.

    (d) The 10th embedding in the second layer of embeddings of a unidirectional Transformer.