Data 188 Introduction to Deep Learning
Spring 2026 Eric Kim

Discussion 06

This discussion covers computer vision and some review!

1. Computer Vision Architecture

(a) Design a two-layer ConvNet image classification model with the following features:

- Input is RGB image with $H = 224, W = 224$.
- Two `Conv2d` layers (with bias), each followed by `BatchNorm2d` and `ReLU`.
- Square filters of size 3×3.
- All intermediate spatial feature maps must have 8 channels.
- Spatial resolution must decrease by a factor of 2 after each convolution.
- Final classification layer for $K = 1000$ classes.

i. Specify the layer parameters (input channels, input features, output channels, output features, stride, padding, etc.) for each layer in the model architecture.

ii. Calculate the total number of trainable parameters in the model architecture you designed in part (i).

(b) Suppose our training batch size B is very small (e.g., $B = 1$). What specific change should we make to the model architecture to ensure training stability?

2. ConvNormAct

Suppose we have the following code:

```
def create_conv_norm_act(C: int) -> Module:
   return Sequential(
      Conv2d(
         in_channels=C,
         out_channels=C,
         kernel_size=3,
         stride=1,
         padding=1,
      ),
      BatchNorm2d(num_features=C),
      ReLU(),
   )
```

Implement the `residual_conv_norm_act_forward` function that adds a "skip connection" to the `ConvNormAct` block. Assume that we use elementwise-addition to implement the skip connection.

```
def residual_conv_norm_act_forward(
   Z_in: Tensor, # shape=[b, c, h, w]
   conv_norm_act: Module, # return value of create_conv_norm_act
) -> Tensor:
   Z_conv = ____________________________________________________
   return ____________________________________________________
```

3. Computational Graph

(a) Draw a computational graph for this single-item, scalar model:

```
# X (input) shape=[1]
# y (ground-truth) shape=[1]
linear1 = Linear(in_feats=1, out_feats=1, bias=False)
relu = ReLU()
linear2 = Linear(in_feats=1, out_feats=1, bias=False)

z1 = linear1(X)
z2 = relu(z1)
z3 = linear2(z2)
y_hat = z2 + z3
L = L2Loss(y_hat, y)
```


(b) Let $X = 1$, and $y = 0.5$. Let the weights of the linear layers be $w_1 = 1$, $w_2 = 2$.

i. Compute the forward pass.

ii. Compute the adjoints for all nodes (do the backward pass).

iii. Compute the final gradients for the trainable parameters w_1 and w_2.

(c) Now consider $X \in \mathbb{R}^2, y \in \mathbb{R}^2, W_1 \in \mathbb{R}^{4\times 2}, W_2 \in \mathbb{R}^{2\times 4}$. This means that we left multiply weight matrices, e.g. $z_1 = W_1 X$ and $z_3 = W_2 z_2$. Additionally, recall:

- Vector L2 loss is defined as $\mathcal{L} = \frac{1}{2}\|\hat{y} - y\|_2^2$
- For the matrix multiplication operation $C = AB$, the gradients are $\frac{\partial \mathcal{L}}{\partial A} = \left(\frac{\partial \mathcal{L}}{\partial C}\right) B^T$ and $\frac{\partial \mathcal{L}}{\partial B} = A^T \left(\frac{\partial \mathcal{L}}{\partial C}\right)$

Express the gradients $\frac{\partial \mathcal{L}}{\partial W_1}$ and $\frac{\partial \mathcal{L}}{\partial W_2}$ symbolically using the upstream gradients.

4. Miscellaneous Review!

(a) Take the Jacobian of the following vector function $\mathbf{f}(x, y, z)$:

$$\mathbf{f}(x, y, z) = \begin{bmatrix} x^2 + y^2 \\ 5z \\ x \sin(y) \end{bmatrix}$$

(b) What are logits? What is the softmax function? What loss do we use in softmax regression?

(c) Recall the equations for the Adam optimizer:

$$u_t = \beta_1 u_{t-1} + (1 - \beta_1)\nabla_\theta L(\theta_t) \quad \text{First moment}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta L(\theta_t))^2 \quad \text{Second moment}$$

$$\hat{u}_t = \frac{u_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad \text{Bias correction}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{u}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad \text{Weight update}$$

What do the first moment, second moment, and bias correction equations do?

(d) Normalize the following matrix using LayerNorm, then BatchNorm. Assume that `X.shape=[batchsize, dim]`. Ignore the learnable parameters of each normalization layer for this question (e.g. no need to scale and shift the normalized output).

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

(e) Given a convolutional layer with a 3×3 kernel and a padding of 2, calculate the output dimensions for an input of size 5×8. Assume a stride of 1.

Contributors:

- Eric Kim, Andria Xu.

