EECS 182 Deep Neural Networks
Spring 2023 Anant Sahai

Discussion 5

1. Residual Connection

Neural networks learn by optimization with gradients. The deeper a neural network grows, the more likely it will suffer from either vanishing gradients or exploding gradients problem since downstream layers (ie. earlier layers) will be optimized with either exponentially small or large gradients due to chain rule. Exploding gradients can be resolved using **gradient clipping**, whereas tackling vanishing gradients is more tricky. Here we study how skip connections can help.

(a) Vanishing gradient results from taking the product of many gradients with norm less than 1. One reason is due to saturating activation functions such as sigmoid. Recall the form of sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$. **What's the maximum gradient that sigmoid can generate, ie. the maximum of $\frac{d}{dx}\sigma(x)$?** Based on what you computed, **why is ReLU more preferable than sigmoid in deep neural nets**?

(b) One common way to tackle vanishing gradients is by adding residual connections. Fig 1 shows a simplified example of a residual block. Assuming that the weights are affine layer with zero biases, **what's the expression of X_1 in terms of W_1, W_2 and X_0?** $W_i \in \mathbb{R}^{n\times n}$, $X_i \in \mathbb{R}^n$.

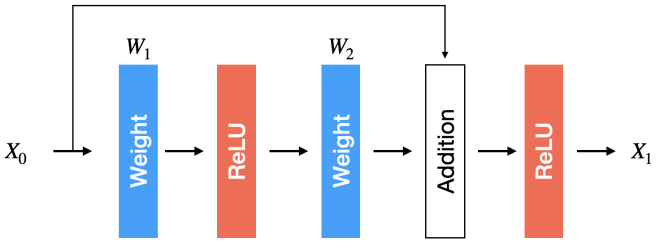

Figure 1: Diagram for Residual Block

(c) **Compute the gradient $\frac{\partial X_1}{\partial X_0}$**. Based on what you see numerically, **why does residual connection preserves gradient norms better**?

2. Regularization and Dropout

Recall that linear regression optimizes the following learning objective:

$$\mathcal{L}(\mathbf{w}) = ||\mathbf{y} - X\mathbf{w}||_2^2 \tag{1}$$

One way of using **dropout** during SGD on the d-dimensional input features $\mathbf{x}_i$ involves keeping each feature at random $\sim_{i.i.d}$ $Bernoulli(p)$ (and zeroing it out if not kept) and then performing a SGD step.

It turns out that such dropout makes our learning objective effectively become

$$\mathcal{L}(\check{\mathbf{w}}) = E_{R\sim Bernoulli(p)}\left[||\mathbf{y} - (R \odot X)\check{\mathbf{w}}||_2^2\right] \tag{2}$$

where $\odot$ is the element-wise product and the random binary matrix $R \in \{0,1\}^{n\times d}$ is such that $R_{i,j} \sim_{i.i.d}$ $Bernoulli(p)$. We use $\check{\mathbf{w}}$ to remind you that this is learned by dropout.

Recalling how Tikhonov-regularized (generalized ridge-regression) least-squares problems involve solving:

$$\mathcal{L}(\mathbf{w}) = ||\mathbf{y} - X\mathbf{w}||_2^2 + ||\Gamma\mathbf{w}||_2^2 \tag{3}$$

for some suitable matrix Γ, it turns out we can manipulate (2) to eliminate the expectations and get:

$$\mathcal{L}(\check{\mathbf{w}}) = ||\mathbf{y} - pX\check{\mathbf{w}}||_2^2 + p(1-p)||\check{\Gamma}\check{\mathbf{w}}||_2^2 \tag{4}$$

with $\check{\Gamma}$ being a diagonal matrix whose j-th diagonal entry is the norm of the j-th column of the training matrix X.

(a) **How should we transform the $\check{\mathbf{w}}$ we learn using (4) (i.e. with dropout) to get something that looks a solution to the traditionally regularized problem (3)?**
(Hint: This is related to how we adjust weights learned using dropout training for using them at inference time. PyTorch by default does this adjustment during training itself, but here, we are doing dropout slightly differently with no adjustments during training.)

(b) With the understanding that the Γ in (3) is an invertible matrix, **change variables in (3) to make the problem look like classical ridge regression:**

$$\mathcal{L}(\widetilde{\mathbf{w}}) = ||\mathbf{y} - \widetilde{X}\widetilde{\mathbf{w}}||_2^2 + \lambda||\widetilde{\mathbf{w}}||_2^2 \tag{5}$$

Explicitly, what is the changed data matrix $\widetilde{X}$ in terms of the original data matrix X and Γ?

(c) Continuing the previous part, with the further understanding that Γ is a *diagonal* invertible matrix with the j-th diagonal entry proportional to the norm of the j-th column in X, **what can you say about the norms of the columns of the effective training matrix** $\widetilde{X}$ and **speculate briefly on the relationship between dropout and batch-normalization.**

Contributors:

- Kevin Li .
- Saagar Sanghavi.
- Jerome Quenum.
- Anant Sahai.

