EECS 182 Deep Neural Networks
Fall 2025 Anant Sahai and Gireeja Ranade

Discussion 3

1. Optimizers as Penalized Linear Improvement

In lecture, you saw the locally linear perspective of a neural network and the loss by Taylor expanding the loss around the current value of the parameters. This approximation is only very good in a near neighborhood of those values. One way to proceed with optimization is to consider the size of the neighborhood as a hyperparameter and to bound our update to stay within that neighborhood while minimizing our linear approximation to the loss. You saw in lecture that the choice of norm in defining that neighborhood also matters.

In this problem (and the homework), you will work out for yourself a slightly different perspective. Instead of treating the norm as a constraint (with the size of the acceptable norm as a hyperparameter), we can do an unconstrained optimization with a weighted penalty that corresponds to the squared norm — where that weight is a hyperparameter.

At each iteration, we wish to maximize linear improvement of the objective (as defined by the dot-product between the gradient and the update) locally regularized by a penalty on the size of the update. This can be expressed (in traditional minimization form) as:

$$u = \underset{\Delta\theta}{\operatorname{argmin}} \quad \underbrace{g^T \Delta\theta}_{\text{Linear Improvement}} \quad + \quad \frac{1}{\alpha} \quad \underbrace{d(\Delta\theta)}_{\text{Distance Penalty}} \quad , \tag{1}$$

where $g = \nabla f(\theta)$ is the gradient of the loss, α is a scalar, and d is a scalar-output distance function $\mathbb{R}^{dim(\theta)} \to \mathbb{R}^+$.

Let's assume *Euclidean distance* is the norm that captures our sense of relevant neighborhoods in parameter space. Then we can be interested in:

$$u = \underset{\Delta\theta}{\operatorname{argmin}} \; g^T \Delta\theta \; + \frac{1}{\alpha} \|\Delta\theta\|_2^2. \tag{2}$$

What is the analytical solution for u in the above problem? What standard optimizer does this recover?

2. RMS Norm

We have learned how SGD and Adam can be seen as constrained optimization problems, defined by some norm over parameter space. In this discussion, we will explore how it is helpful to have suitable norms over the *output features* of a neural net layer.

(a) The Euclidean norm is often utilized to define the "scale" of a feature vector:

$$||\mathbf{x}||_2 = \sqrt{\sum_i x_i^2}.$$

However, the *root-mean-squared* (RMS) norm:

$$||\mathbf{x}||_{RMS} = \sqrt{\frac{1}{d}\sum_i x_i^2},$$

is often more appropriate when we care about the average scales of *individual feature elements*.

How do the two norms scale with dimension d (i.e. the number of elements in x), assuming each element is sampled from a standard Gaussian distribution?

(b) **For a vector space with dimension $d = 2$, draw out the set of points with distance 1 under each norm below.**

- The Euclidean norm (or 2-norm):
$$||\mathbf{x}||_2 = \sqrt{\sum_i x_i^2}.$$
- The RMS Norm:
$$||\mathbf{x}||_{RMS} = \sqrt{\frac{1}{d}\sum_i x_i^2}.$$
- The 1-Norm:
$$||\mathbf{x}||_1 = \sum_i |x_i|.$$
- The Infinity Norm:
$$||\mathbf{x}||_\infty = \max_i |x_i|.$$

(c) Consider a neural net layer with input $\mathbf{x} \in \mathbb{R}^{d_1}$, weights $W \in \mathbb{R}^{d_2 \times d_1}$, where W is initialized with i.i.d. standard Gaussian entries, and entries of $\mathbf{x}$ are sampled from i.i.d. unit Gaussians.

What is the expected squared RMS norm of the output features $\mathbf{y} = W\mathbf{x}$? How does this scale with d_1 or d_2? What constant should we multiply W by to ensure that the expected squared RMS norm of $W\mathbf{x}$ is 1, regardless of d_1 and d_2?

Hint: Consider a simplified layer with a single output feature, $W \in \mathbb{R}^{1 \times d_1}$. What is the variance of the scalar $y = W\mathbf{x}$?

3. Optimizers and their convergence

In this question, we will examine how various optimizers converge to different points when there is a manifold of parameter values that all achieve zero training loss.

For this part, we have exactly $n = 1$ training point corresponding to the single equation

$$[1, 0.1, 0.01]\boldsymbol{\theta} = 1 \tag{3}$$

with a 3-dimensional column vector of parameters $\boldsymbol{\theta}$. Suppose that we start with $\boldsymbol{\theta}_0 = \mathbf{0}$ and use squared loss $f_t(\boldsymbol{\theta}) = (1 - [1, 0.1, 0.01]\boldsymbol{\theta})^2$.

- **What specific vector $\boldsymbol{\theta}^*$ would standard vanilla SGD converge to assuming the learning rate was small enough to give convergence?**
- **What specific vector $\boldsymbol{\theta}_2^*$ would signSGD converge to assuming an appropriate learning rate schedule to give convergence?**

Contributors:

- Kevin Frans.
- Anant Sahai.
- Gireeja Ranade.
- Luke Jaffe.
- Kevin Li.

