CSE 446: Machine Learning **Lecture**

Stationary points, non-convex optimization, and more...

Instructor: Sham Kakade

1 Terminology

- stationary point of $f(w)$: a point which has zero gradient.
- local minima of $f(w)$: a point which locally is at a minima (i.e. any infinitesimal change to the point will result in an infinitesimal decrease in the function value).
- global minimum of $f(w)$: a point w_* which achieves the minimal possible value of $f(w)$ over *all* w.
- saddle point of $f(w)$: a point which will go up, under some infinitesimal perturbation, and will go down, under some other infinitesimal perturbation.

Issues related to training are:

- non-convexity
- initialization
- weight symmetries and "symmetry breaking"
- saddle points & local optima & global optima
- vanishing gradients

2 Gradient descent in the non-convex setting

Suppose we do gradient descent on a function F:

$$w^{(k+1)} = w^{(k)} - \eta^{(k)} \cdot \nabla F(w^{(k)}) \,.$$

We could also do SGD:

$$w^{(k+1)} = w^{(k)} - \eta^{(k)} \cdot \widehat{\nabla F(w^{(k)})} \,.$$

where $\widehat{\nabla F(w^{(k)})}$ is some (unbiased) estimate of the gradient. The basic question is: where does this lead us to?

Here, we do *not* assume that F is convex. If Fwere convex, then we would indeed get to the global optima of our function, under mild restrictions.

2.1 Informal convergence statement

In short, by doing GD or SGD (and setting learning rates appropriately), our function value will decrease for a while until we hit a point whose gradient is near to 0. We are not able to say much beyond this: it is (computationally) difficult to determine if we are near to a saddle point, a local minima, or a global minima. And, in practice, it is not obvious which case we are in! It is entirely plausible that we get stuck at saddle points or "plateaus"; sometimes these points might have pretty reasonable function values.

Here is the informal convergence claim for GD: with a constant learning rate (set appropriately), the guarantees are: 1) we will decrease the function value after every update (if our gradient is not zero) 2) in $O(k)$ iterations, we will find a point whose gradient has square norm that is "small", of size $O(1/k)$. In other words, we will get to an approximate stationary point. In general, it is difficult to tell if this point is a saddle point, a local minima, or a global minima.

For SGD with an (appropriately) decaying learning rate, we: 1) will have an (expected) decrease in the function value after every update (provided our gradient is not zero) 2) after k updates, we will find a point whose gradient has square norm that is "small", of size $O(1/\sqrt{k})$. Again, we will get to an approximate stationary point.

2.2 Implications: initialization, weight symmetries, and practical guidance

Initialization:

- Starting with all the weights being 0 is often a saddle point. Even if not, it often forces certain constraints on our weights (See the HW and Bishop).
- Starting weights too large can also be problematic. This is because for some activation functions this is actually a small gradient point.
- If it were me, I like to initialize so the activations to any hidden nodes are unit variance (this can be done just by some scaling tricks). This is sometimes called "Xavier" initialization. Basically, I like to think about the scalings in terms of the variance of a hidden unit when I start things off.

When are gradients near to 0?

- If all our ReLu units turn off, then we are at a stationary point. This is clearly not desirable.
- If we "saturate" our sigmoid or tanh units, then the gradient is also near to 0. Sometimes this is problematic, since if our learning rate is very "large" then our transfer function saturates quickly and this could slow the learning process down.

Symmetries:

- Read Bishop and CIML
- See CIML and understand how you can "swap" two hidden units in a one layer hidden network.
- do the homework

2.3 More formal statements

This is not required reading. If you are interested, these are more precise statements. In one of the extra credit problems, you can prove one of our claims.

Let us assume the that our function F is "smooth" (see the A4.pdf for a definition). Basically, this is a mild regularity condition saying that gradients are stable to small perturbations in their inputs.

Theorem 2.1. *(Gradient descent case) If we use a constant learning rate (and set the learning rate appropriately), then:*

- *Gradient descent will* always *decrease the function value at every step (as long as gradient is not exactly* 0*).*
- *After* k *updates, we will be guaranteed to have found some point which has a gradient whose square norm is less than* $O(1/k)$*. Precisely, we will find some* $w^{(k')}$*, with* $k' \leq k$*, in which* $\|\nabla F(w^{(k')})\|^2$ *is* $O(1/k)$*.*

(Stochastic gradient descent case) If we use an appropriately decaying *learning rate, then:*

- *SGD, in* expectation*, decreases the function value at every step (as long as gradient is not exactly* 0*).*
- *After* k *updates, we will be guaranteed to have found some point which has a gradient whose square norm is less than* $O(1/\sqrt{k})$*.*