Stochastic Optimization for Machine Learning Model Training*

Caleb Manicke

1 Introduction

We live in an age of unprecedented technological sophistication, where cars can drive themselves, and chat models can do homework for students. The largest headlines have all been in machine learning: by combining intuitions across several fields of mathematics, most notably optimization, probability, and statistics, machine learning models have made great strive in their capability. One of the most important fields that have contributed to machine learning is stochastic processes. Stochastic processes have allowed researchers and developers to find a sense of certainty in training models despite an element of randomness always being present. The purpose of this paper is to give a formal introduction, rigorous proof, and working intuition behind the most popular stochastic method in machine learning training: Stochastic Gradient Descent.

2 Formal Definitions and Objective

Before we uncover how stochastic processes have helped optimize machine learning model training, we need to formally define what it is we are optimizing.

Definition 1 (Dataset of Images). *We define a data set as a finitely countable set of points* $\{x_i\}_{i=1}^n$. *We will work with images, so each point* x_i *in our dataset is an image with* d_x *pixels defined as a row vector* $x_i = (x_{i,1}, ..., x_{i,d_x}) \in \mathbb{R}^{d_x}$.

Definition 2 (Labels). *We define our labels as a finitely countable set of integers* $\{y_i\}_{i=1}^n$. *Each* y_i *denotes a class label for its corresponding image* x_i.

Example - Suppose our dataset of images fit into 10 categories, {Cat, Dog, Plane, Truck, ...}, so each x_i is an *image* of an object in one of these classes. Then the label of x_i, denoted y_i, should be the integer which denotes which class x_i belongs to, so if x_3 is a picture of an airplane, then $y_i = 3$.

*Source: `https://www.overleaf.com/project/63d07d57e3c07006b484dc9d`

Definition 3 (Training Set). *Given our images 1 and our labels 2 as define above, our training set is the set of pairs of each image and their corresponding label, so* $\{(x_1, y_1), (x_2, y_2), ...\}$.

Suppose we have a training set of images and labels. A new machine learning model would use this training set in order to find parameters that would allow it to accurately choose labels for new images. This is our optimization problem: we want to find a classifier, i.e. a model that picks a label for an input image, that minimizes the number of times it mislabels an image. [4]

Definition 4 (Classifier). *We express our classifier as a function that maps an image* $x_i \in \mathbb{R}^{d_x}$ *to a label* $y_i = \{1, 2, 3, ...\}$, *so* $h : \mathbb{R}^{d_x} \to \{1, 2, 3, ...\}$.

Definition 5 (Classifier Objective). *We want to find a classifier* h *such that* $h(x_i) = y_i$ *for most* i. *Formally, we want to solve* $\min_n \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$, *i.e. minimize the number of examples our classifier's label is wrong. Recall our indicator function* $\mathbb{1}$ *takes events as input, so* $\mathbb{1}(A) = 1$ *if* A *is true and* $\mathbb{1}(A) = 0$ *otherwise.*

This is our formal optimization problem. However, we need to make some tweaks to this: if we want to optimize the number of examples our classifier gets right, then we need to find the right *parameters* for our classifier. Our problem is exactly like trying to find the best fit-line in a plane: if $h(x) = ax + b$ is our best fit line, then we want to find **weights** $w = \{a, b\}$ such that for each example x_i, $h(x_i)$ is as close to each y_i as possible.

Another problem is that since $\mathbb{1}$ only returns either 1 or 0, our minimization function only returns discrete values. This is not the best for optimization problems: it's much easier to optimize a continuous and differentiable function since we can use its derivative to update its weights. We define a continuous **loss** function $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ that approximates $\mathbb{1}$, so we can redefine our classifier objective as follows:

Definition 6 (Classifier Objective). *We want to solve* $min_{w \in \mathbb{R}^{d_x}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(x_i; w), y_i)$, *i.e. we want to find the best weights* $w \in \mathbb{R}^{d_x}$ *such that for each example image* x_i, *the value our loss function returns after comparing our model's output with the correct label is minimally small.*

3 Stochastic Gradient Descent

Now that we have established what our main goal is, we can start discussing how we can achieve this. We need an algorithm that will find our optimal parameters, so this is where we introduce **Stochastic Gradient Descent**.

Algorithm 1 **Stochastic Gradient Descent**

1. Receive initial point w_1, max iterations $T \in \mathbb{N}$, and step sizes $\{a_k\}_{k=1}^{T}$
2. For $k = 1, ..., T$ do:
3. Choose a random image x and the k-th step size a_k
4. Update $w_{k+1} = w_k - \alpha_k \Delta h(x; w_k)$

Before it can be explained what Stochastic Gradient Descent does, we must first intuitively define what *space* it works in. We can imagine our Parameter $\times$ Loss space as a bowl in $\mathbb{R}^{d_x} \times \mathbb{R}$ dimensions, where each possible set of parameters in our parameter space $\mathbb{R}^{d_x}$ maps to an "accuracy" in the loss space in $\mathbb{R}$, i.e. its score in our optimization problem $\mathcal{L}(h(x_i; w), y_i)$. Because we are dealing with a "bowl", we imagine there is an "optimal" set of parameters with the lowest possible loss that exists in the middle of our parameter space.

We can find this "optimal" parameter point by descending *against* the direction of our score's gradient. This is what Stochastic Gradient Descent does: at each step, we pick a completely random image from our training set and "step" size. We descend down our bowl by the amount specified by our step size as well as by how accurate our classifier predicted that random image.

Because we update our parameters after every example, our loss function between each iteration is bound to act sporadically, moving up and down our bowl in a seemingly incoherent manner. However, as our parameters become more adjusted, there is a general trend where our model oscillates around the "optimal" set of parameters in our middle of our bowl before converging.

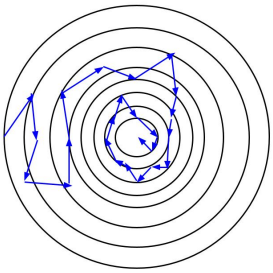

Figure 1: For the sake of intuition, image a parameter space in two-dimensions, where each ring in our diagram above represents a cross-section of our bowl. SGD oscilates in any random direction around our convex bowl, but will eventually converge to our optimal minimum and stay there in the long run.

4 Proving Stochastic Gradient Descent Works

We mention above that Stochastic Gradient Descent will converge to our "optimal" set of parameters, but how can we know for certainity that this will happen? In order to explain this, we need to further formalize our objectives and prove that our seemingly random process abides by certain trends. This will also give us leeway into our intuition of SGD as a stochastic process.

Definition 7 (Stationary Point). *A point w^* is a stationary point for some function F if $\Delta F(w^*) = 0$.*

Definition 8 (Euclidean Norm). *For $x = (x_1, ..., x_n)$ we define $||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$.*

Definition 9 (Random Seed). *Let $\mathcal{E}_i$ denote the random variable, or seed, which chooses our datapoint at the i-th step.*

We will work with a new, continuously differentiable functions defined on random seeds. Denote $f = \mathcal{L} \circ h$ as the composition function of our loss and classifier, such that $f_i(w) = f(w, \mathcal{E}_i) = \mathcal{L}(h(x_i; w), y_i)$ is the loss at the i-th step given the i-th randomly chosen image. We hope to minimize the expected loss at each step, which we denote as $F(w) = \mathbb{E}[f(w, \mathcal{E})]$.

As our number of iterations $T \to \infty$, we want to show that $\Delta F(w) \to 0$, that is, our expected loss will have found a stationary point which is our minimum. In order to show that our change in expected loss converges, we will show that our losses and their derivatives are bounded through several theorems. To show boundedness, we need our expected change in loss to have a strong continuous property, so we assume it is Lipschitz continuous.

Definition 10 (Lipschitz Continuity). *[3] Define $F : A \to \mathbb{R}$ where $A \subseteq \mathbb{R}^n$. If F is Lipschitz continuous, then for all $x, y \in A$, there exists a constant $L \in \mathbb{R}$ such that $||F(x) - F(y)||_2 \leq L \cdot ||x - y||_2$.*

Since the gradient of the expected loss $\Delta F(w)$ is its own differentiably continuous function, we assume it's Lipschitz continuous. Our first theorem shows that if this is true, we can derive a useful inequality about the bounds for our expected loss with relation to any two weights.

Theorem 1 (Bound of Loss). *[1]*

Suppose $F : \mathbb{R}^{d_x} \to \mathbb{R}$ is continuously differentiable and $\Delta F : \mathbb{R}^{d_x} \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L such that $||\Delta F(w) - \Delta F(v)||_2 \leq L \cdot ||w - v||_2$ for all $w, v \in \mathbb{R}^{d_x}$.

Then $F : \mathbb{R}^{d_x} \to \mathbb{R}$ satisfies $F(w) \leq F(v) + \Delta F(v)^T (w - v) + \frac{1}{2} L ||w - v||_2^2$ for all $w, v \in \mathbb{R}^{d_x}$ (where $F(v)^T$ is a transpose matrix).

Proof.

- We can define $F(w) - F(v)$ as an integral from 0 to 1 over dt of the derivative of F on the line between w and v, so $F(w) - F(v) = \int_0^1 \frac{\partial F(v + t(w - v))}{\partial t} dt$. Rearranging these terms gives us $F(w) = F(v) + \int_0^1 \frac{\partial F(v + t(w - v))}{\partial t} dt$.

- By Chain Rule for functions of several variables, if we differentiate $\frac{\partial F(v+t(w-v))}{\partial t}$ inside our integral, we get $F(w) = F(v) + \int_0^1 \Delta F(v+t(w-v))^T (w-v) dt$.
- Consider the term $\Delta F(v)^T \cdot (w-v)$. This is an inner-product, so we can add this term alongside its negation as follows: $F(w) = F(v) + [\Delta F(v)^T \cdot (w-v) - \Delta F(v)^T \cdot (w-v)] + \int_0^1 \Delta F(v+t(w-v))^T (w-v) dt = F(v) + \Delta F(v)^T \cdot (w-v) + \int_0^1 [\Delta F(v+t(w-v))^T - \Delta F(v)^T](w-v) dt = F(v) + \Delta F(v)^T \cdot (w-v) + \int_0^1 [\Delta F(v+t(w-v)) - \Delta F(v)]^T (w-v) dt$.
- We apply a Cauchy-Schwartz inequality $|\langle a, b \rangle| = a^T b \leq ||a||_2 \cdot ||b||_2$. If we apply it to our dot-product inside our integral, we get that $F(w) \leq F(v) + \Delta F(v)^T (w-v) + \int_0^1 ||\Delta F(v+t(w-v)) - \Delta F(v)||_2 \cdot ||w-v||_2 dt$.
- We can apply the definition of Lipschitz continuity on our norm $||\Delta F(v+t(w-v)) - \Delta F(v)||_2$, which gives us $F(w) \leq F(v) + \Delta F(v)^T (w-v) + \int_0^1 L||t(w-v)||_2 \cdot ||w-v||_2 dt$.
- Taking the t out of our norm in our integral will give us our final expression $F(w) \leq F(v) + \Delta F(v)^T (w-v) + \frac{1}{2} L||w-v||_2^2$.

□

We offer an intuitive approach: by the definition of Lipschitz continuity, the gradients for the expected change in loss of two weights w, v is bounded by a constant L, formally written as $||\Delta F(w) - \Delta F(v)||_2 \leq L \cdot ||w-v||_2$. When re-arranged, our inequality above gives us $F(w) - F(v) \leq \Delta F(v)^T (w-v) + \frac{1}{2} L||w-v||_2^2$. Thus, our theorem tells us that if the expected change in loss of two weights is Lipschitz continuous, then their losses are bounded by the same L constant. This is why we refer to this theorem as "Bound of Loss".

Although our theorem 1 above gives us a clear notion of how the bounds of loss and change of loss are related, we need to be most specific and define our change of loss in terms of what random example we used to find our loss on, as found by our "seed".

Definition 11 (Expectation of Seed). *Let $\mathbb{E}_{\mathcal{E}_k}[X]$ be the expected value taken with respect to the distribution of the random "seed" variable $\mathcal{E}_k$ that samples our k-th example, given some variable X.*

We consider the expected change of loss $F(w_{k+1})$ at our $k+1$-th step given our seed $\mathcal{E}_k$. We want to bound this term $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})]$ by some quantity defined by our *previous* change of loss, our loss for our previous weight w_k, and our step size α_k to affirm that no matter what example our seed chooses, our weights will change *no greater* than our decreasing quantities α_k and $\Delta f(w_k, \mathcal{E}_k)$.

Theorem 2 (Bound of Expectation). *[1] Assume $F : \mathbb{R}^{d_x} \to \mathbb{R}$ is continuously differentiable and $\Delta F : \mathbb{R}^{d_x} \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L. Then $\mathbb{E}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \Delta F(w_k)^T \mathbb{E}[\Delta f(w_k, \mathcal{E}_k)] + \frac{1}{2}\alpha_k^2 \cdot L \cdot \mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2]$ for all $k = 1, ..., T$.*

Proof.

- Our assumptions for this theorem match those of our previous theorem 1, so for two points $w, v \in \mathbb{R}^{d_x}$, we have $F(w) \leq F(v) + \Delta F(v)^T(w - v) + \frac{1}{2}L||w - v||_2^2$.
- If we replace w and v with two consecutive weights w_{k+1} and w_k respectively, we get $F(w_{k+1}) - F(w_k) \leq \Delta F(w_k)^T(w_{k+1} - w_k) + \frac{L}{2}||w_{k+1} - w_k||_2^2$.
- Recall our stochastic gradient descent algorithm 1, where our update rule sets $w_{k+1} = w_l - \alpha_k \Delta h(x_k; w_k)$ where $\Delta h(x_k; w_k) = \Delta f(w_k; \mathcal{E}_k)$.
- If we replace all qualities $w_{k+1} - w_k$ in our inequality with $-\alpha_k \Delta f(w_k; \mathcal{E}_k)$, we get $F(w_{k+1}) - F(w_k) \leq -\alpha_k \Delta F(w_k)^T \cdot \Delta f(w_k; \mathcal{E}_k) + \alpha_k \cdot \frac{L}{2} \cdot ||\Delta f(w_k; \mathcal{E}_k)||_2^2$.
- Now we will take the expectation of each term in our equality with respect to $\mathcal{E}_k$. Recall that we use $\mathcal{E}_k$ to randomly choose a point **at** the k-th step, but we did not use it to determine our quantities at the k-th step. This means our expected change of loss $F(w_k)$ is independent from $\mathcal{E}_k$, but our next weights w_{k+1} were chosen using $\mathcal{E}_k$, so $F(w_{k+1})$ and the change of loss after choosing our example $\Delta f(w_k, \mathcal{E}_k)$ are dependent on $\mathcal{E}_k$.
- Thus, when we take expectations with respect to $\mathcal{E}_k$, we get $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \Delta F(w_k)^T \mathbb{E}[\Delta f(w_k, \mathcal{E}_k)] + \frac{1}{2} \cdot \alpha_k^2 L \cdot \mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2]$.

□

Although we try to be as precise as possible by taking expectations and gradients with respect to our random seed $\mathcal{E}_k$, there will always be some "noise" from our random process. This means if we want to create more precise boundaries for our weights between each iteration of SGD, we need to include a general metric for our "noise". In our next theorem, we will bound our expected change of loss *after* choosing our k-th example *with respect* to our k-th seed $\mathcal{E}_k$, formally written as $\mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2]$, by constants M and M_G that represent the "noise" of our gradients that are independent from our process.

Theorem 3 (Bound of Iteration). *[2] Suppose there exist constants $M \geq 0$ and $M_G \geq 1$ such that $\mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2] \leq M + M_G||\Delta F(w_k)||_2^2$. Furthermore, assume $F : \mathbb{R}^{d_x} \to \mathbb{R}$ is continuously differentiable and $\Delta F : \mathbb{R}^{d_x} \to \mathbb{R}$ is Lipschitz continuous.*

Then the iterations of SGD satisfy $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -(1 - \frac{1}{2}\alpha_k L M_G)\alpha_k ||\Delta F(w_k)||_2^2 + \frac{1}{2}\alpha_k L M$ for all $k = 1, ..., T$.

Proof.

- Our suppositions match those of our previous theorem 2. Therefore, by this theorem, we get $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \Delta F(w_k)^T \mathbb{E}_{\mathcal{E}_k}[\Delta f(w_k, \mathcal{E}_k)] + \frac{1}{2}\alpha_k^2 \cdot L \cdot \mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2]$.
- At our k-th seed $\mathcal{E}_k$, our expected gradient or change of loss $\mathbb{E}_{\mathcal{E}_k}[\Delta f(w_k, \mathcal{E}_k)]$ at a single datapoint x_i is just $\Delta F(w_k)$, so replacing this term gives us with $F(w_k)$ gives us $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \Delta F(w_k)^T F(w_k) + \frac{1}{2}\alpha_k^2 \cdot L \cdot F(w_k)$
- Since $F(w_k)^T \cdot F(w_k)$ is a dot product of the same term, it is equal to the squared Euclidean norm of $F(w_k)$, so $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k ||F(w_k)||_2^2 + \frac{1}{2}\alpha_k^2 \cdot L \cdot \mathbb{E}_{\mathcal{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2]$.
- Plugging in our bound noise assumption for $\mathbb{E}_{\mathcal{E}_k}[\Delta f(w_k, \mathcal{E}_k)]$ gives us $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k ||F(w_k)||_2^2 + \frac{1}{2}\alpha_k^2 L(M + M_G ||\Delta F(w_k)||_2^2)$. Simplifying this gives us our final statement $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -(1 - \frac{1}{2}\alpha_k L M_G)\alpha_k ||\Delta F(w_k)||_2^2 + \frac{1}{2}\alpha_k L M$.

□

We have established solid ground for how our weights are bounded according to the gradients of our loss function, step sizes, and even independent noise. With this we can explore the convergence of our Stochastic Gradient Descent process. Recall that we want to find a stationary point such that $\Delta F(w_k) = 0$ after some number of iterations k. Our next algorithm shows that when we have a converging sequence of diminishing step sizes, so $\sum_{i=1}^{\infty} a_k = \infty$ but $\sum_{i=1}^{\infty} a_k^2 < \infty$, as $T \to \infty$, $\Delta F(w_k) \to 0$.

Theorem 4 (Convergence of Gradients). *[2] Suppose there exist constants $M \geq 0$ and $M_G \geq 1$ such that $\mathbb{E}_{\mathbb{E}_k}[||\Delta f(w_k, \mathcal{E}_k)||_2^2] \leq M + M_G ||\Delta F(w_k)||_2^2$. Furthermore, assume $F : \mathbb{R}^{d_x} \to \mathbb{R}$ is continuously differentiable and $\Delta F : \mathbb{R}^{d_x} \to \mathbb{R}$ is Lipschitz continuous.*

Suppose we have a diminishing step size sequence such that $\sum_{i=1}^{\infty} a_k = \infty$ but $\sum_{i=1}^{\infty} a_k^2 < \infty$. Define $A_T = \sum_{i=1}^{T} a_k$. Then $\lim_{T\to\infty} \mathbb{E}\left[\sum_{k=1}^{T} a_k ||\Delta F(w_k)||_2^2\right] < \infty$ and therefore $\mathbb{E}\left[\frac{1}{A_T}\sum_{k=1}^{T} a_k \cdot ||\Delta F(w_k)||_2^2\right] \to 0$ as $T \to \infty$. [2]

Proof.

- The assumptions we make for this theorem match those in our previous theorem 3. Therefore, we know that $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -(1 - \frac{1}{2}\alpha_k L M_G)\alpha_k ||\Delta F(w_k)||_2^2 + \frac{1}{2}\alpha_k L M$.
- We know that $k \to \infty$, $\alpha_k \to 0$, so without loss of generality for our k we can assume that $a_k \leq \frac{1}{LM_G}$ (since α_k becomes arbitrarily small). This inequality allows us to simplify our term $-(1 - \frac{1}{2}\alpha_k L M_G) \leq \frac{-1}{2}$, so $\mathbb{E}_{\mathcal{E}_k}[F(w_{k+1})] - F(w_k) \leq -\frac{1}{2}\alpha_k ||\Delta F(w_k)||_2^2 + \frac{1}{2}\alpha_k L M$.

- Denote $F_{\text{inf}} = F(w_T)$ as a constant where $T \to \infty$. When we sum over every inequality above for each iteration $k = 1, ..., T$, we get $F_{\text{inf}} - F(w_1) \leq \mathbb{E}[F(w_T)] - F(w_1) \leq -\frac{1}{2}\sum_{k=1}^{T} a_k \mathbb{E}[||\Delta F(w_k)||_2^2] + \frac{1}{2}LM\sum_{k=1}^{T} a_k^2$.
- We can simplify this inequality by multiplying both sides of our inequality by 2, moving our sum inside our expectation, and moving our expectation to the left-hand side. This will give us $\mathbb{E}\left[\sum_{k=1}^{T} ||\Delta F(w_k)||_2^2\right] \leq 2(F(w_1) - F_\infty) + LM\sum_{k=1}^{T} \alpha_k^2$.
- Dividing both sides by $\frac{1}{A_T}$ gives us our total equation $\dfrac{\mathbb{E}\left[\sum_{k=1}^{T} ||\Delta F(w_k)||_2^2\right]}{A_T} \leq \dfrac{2(F(w_1) - F_\infty) + LM\sum_{k=1}^{T} \alpha_k^2}{A_T}$.
- Since as $T \to \infty$, $A_T \to \infty$, we can conclude that $\dfrac{\mathbb{E}\left[\sum_{k=1}^{T} ||\Delta F(w_k)||_2^2\right]}{A_T} \leq \dfrac{2(F(w_1) - F_\infty) + LM\sum_{k=1}^{T} \alpha_k^2}{A_T} \to 0$ (since $F(w_1) - F_\infty$, L, M are constants, and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, our numerator term is bounded by ∞).

□

Our theorem shows us that as $T \to \infty$, the squared norm of the change of expected loss $||\Delta F(w_k)||_2^2$ will converge to 0. This means our norm for the change of expected loss becomes so small that even though our step sizes α_k also approaches 0, the ratio of loss to step size also approaches 0, so the change to weight made at our k-th step becomes so minuscule it might as well have left our previous weights untouched. We have shown that with SGD, our change to weight will reach 0, which can only happen at a stationary point. Therefore, we can conclude that we are bound to find our optimal minimum as long as our number of iterations is large enough.

5 Stochastic Process Formal Interpretation

We have proven that our algorithm finds a stationary minimum when we train on independent, identically distributed random variable "seeds" for picking our next data point. In our last section we will formalize Stochastic Gradient Descent as a stochastic process defined on a clear state space.

Our state space is the plane of every possible parameter combination. We represent our input images as vectors of $\mathbb{R}^{d_x}$ pixel values, so our weights will also be defined in $\mathbb{R}^{d_x}$. Define a process $(X_t)_{t \geq 0}$ such that each X_t is a random variable that represents our weights at time t, our initial weight X_0 is a randomly chosen set of starting parameters, and $X_{t+1} = X_t - \alpha_t \Delta h(x; X_t)$ where α_t is the t-th step-size and x is a randomly chosen image.

We are interested only in the long term behavior of our process, and we have painstakenly proven that as $t \to \infty$, $\Delta h(x; X_i) \to 0$ for any image x. Since our state space is the entirety of $\mathbb{R}^{d_x}$, our long-term transition matrix $\lim_{n\to\infty} \mathcal{P}$ will be entirely empty in the row where our optimal minimum $(a, b, c, ...) \in \mathbb{R}^{d_x}$ is located in, so $\lim_{n\to\infty} p^n_{(a,b,c,...),(a,b,c,...)} = P_{\text{Start from optimal minimum}}(\text{Return to optimal minimum starting in 1 step}) = 1$ since we won't be able to change our parameters once we reach the stationary point.

Every row in our transition matrix has to add up to 1. To explain this, we employ some intuition: imagine that the closer a point on a parabola is to the vertex, the smaller the gradient of its tangent line will be. This means if our points x on a parabola could take a step of the size of its derivative to another point on the parabola, the closer it is to the vertex, the more likely it will still stay relatively close to the vertex. The same intuition holds for our long-term transition matrix: the closer the entries are to our optimal minimum row-wise, the less spread our their probabilities will be, so our non-zero entries will be concentrated more closer together while being centered around the optimal minimum.

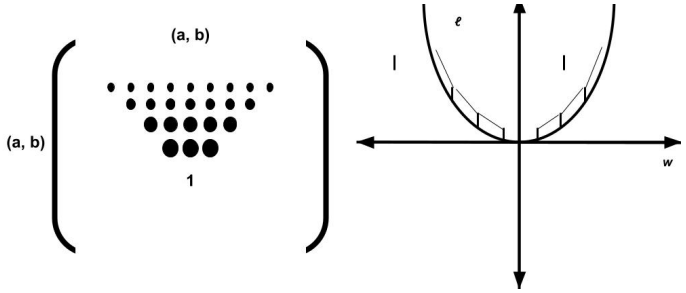

Figure 2: Consider a linear classifier $h(x) = ax + b$ in $\mathbb{R}^2$, and suppose we find some optimal parameter pair (a, b) that minimizes our loss. Then the only entry that is 1 in our long-term transition matrix is our optimal minimum. Furthermore, the closer our probabilities are to our optimal minimum, the larger and more concentrated they will be. This looks similar to how the gradients for the tangent line of points in a parabola are smaller when those points are closer to our vertex. Therefore, once we descend down our slope, it's much less probable our current weight can climb back up since it can't take larger steps.

We can split apart our state space into several classes. The most obvious example is our optimal minimum: once we reach our stationary point, our weights cannot change in future iterations, so we are bound to return to the same set of parameters for all future steps. Therefore, our optimal minimum represents a positive-recurrent class of only one entry. Assuming there are no other stationary points, because we have proved that we expect our process to abandon every state that is not our optimal minimum as our number of iterations $T \to \infty$, the rest of our state space is our transient class.

References

[1] Özgür Martin. Introduction to stochastic optimization for large-scale machine learning part 1. https://www.youtube.com/watch?v=Q_DahzeZhLs.

[2] Özgür Martin. Introduction to stochastic optimization for large-scale machine learning part 2. https://www.youtube.com/watch?v=fBeOV_xrmaI.

[3] Andrew McCrady. Lipschitz functions: Intro and simple explanation for usefulness in machine learning, Jul 2021. https://www.youtube.com/watch?v=UjvFFXakMks.

[4] Mariana Oliveira Prazeres. Stochastic gradient descent: An intuitive proof, Dec 2020.