

Tópicos Avanzados en Entrenamiento de Redes Profundas

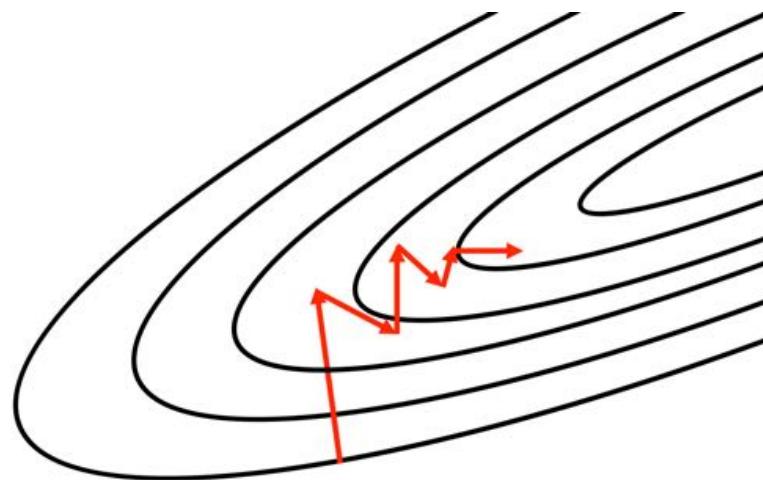
Convergencia de BP y Moralejas



Prof. Ricardo Ñanculef - Departamento de Informática UTSMS

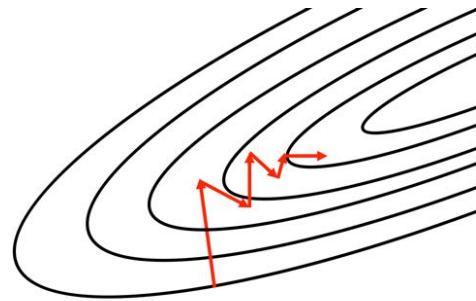
Objetivos

- Demostrar la convergencia de BP estocástico y no estocástico.
- Ilustrar la influencia de la tasa en el resultado.
- Ilustrar la influencia de la “suavidad” de la función en el resultado.
- Ilustrar la influencia de la “estabilidad” de nuestras aproximaciones al gradiente.
- Mencionar técnicas simples y populares de control de la tasa.



Convergencia de BP No Estocástico

- Forma Básica de Método:



- Problema: $\min_w E(\mathbf{w})$
- Algoritmo:
1 **for** $t = 1, \dots, T$ **do**
2 | $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} - \eta \frac{\delta E}{\delta \mathbf{w}};$
3 **end**

- Motivación:

$$1 \quad E(\mathbf{v}) \approx E(\mathbf{w}) + \nabla_{\mathbf{w}} E^T (\mathbf{v} - \mathbf{w})$$

$$2 \quad \nabla_{\mathbf{w}} E^T (\mathbf{v} - \mathbf{w}) < 0 \Rightarrow E(\mathbf{v}) < E(\mathbf{w})$$

$$3 \quad \mathbf{v} - \mathbf{w} = -\nabla_{\mathbf{w}} E \Rightarrow \nabla_{\mathbf{w}} E^T (\mathbf{v} - \mathbf{w}) = -\|\nabla_{\mathbf{w}} E\|^2$$

Convergencia de BP No Estocástico

Teorema

Si $E(\mathbf{w})$ es suave, no degenerada y η es suficientemente pequeña, el método anterior converge a un punto estacionario.

Punto Estacionario

$$\nabla_{\mathbf{w}} E = \frac{\delta E(\mathbf{w})}{\delta \mathbf{w}} = \mathbf{0}$$

Suavidad

$$\|\nabla E(\mathbf{w}) - \nabla E(\mathbf{v})\| \leq L \|\mathbf{w} - \mathbf{v}\|$$



Working Plan

Una iteración del algoritmo tiene la forma:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_t \nabla E(\mathbf{w}^{(t)})$$

Nos gustaría mostrar que $E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)})$ (mejora obtenida en una iteración) es al menos una “fracción” del gradiente, de modo que el algoritmo no se estanca sino es después de encontrar un punto crítico.

Usaremos la suavidad de gradiente

$$\|\nabla E(\mathbf{w}) - \nabla E(\mathbf{v})\| \leq L \|\mathbf{w} - \mathbf{v}\|$$

para obtener una cota de $E(\mathbf{v}) - E(\mathbf{w}) - \langle \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle$:

Al aplicarla a dos iteraciones sucesivas $w = \mathbf{w}^{(t)}, v = \mathbf{w}^{(t+1)}$ obtendremos lo que buscábamos.



Suavidad & Cota para la Mejora

Notemos que:

$$\frac{\partial E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w}))}{\partial \tau} = \langle \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})), (\mathbf{v} - \mathbf{w}) \rangle .$$

De modo que:

$$\int_0^1 \langle \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})), (\mathbf{v} - \mathbf{w}) \rangle d\tau = \int_0^1 \partial E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})) d\tau = (E(\mathbf{v}) - E(\mathbf{w})) .$$

$$E(\mathbf{v}) - E(\mathbf{w}) - \langle \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle =$$

$$\begin{aligned} & \int_0^1 \langle \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})), (\mathbf{v} - \mathbf{w}) \rangle d\tau - \langle \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle \\ &= \int_0^1 \langle \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})) - \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle d\tau \end{aligned}$$



Suavidad & Cota para la Mejora

$$\begin{aligned} E(\mathbf{v}) - E(\mathbf{w}) - \langle \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle &= \int_0^1 \langle \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})) - \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle \, d\tau \\ &\leq \int_0^1 \| \nabla E(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w})) - \nabla E(\mathbf{w}) \| \| (\mathbf{v} - \mathbf{w}) \| \, d\tau \quad \text{Por Cauchy-Swartz} \end{aligned}$$

Por la suavidad del gradiente

$$\begin{aligned} |E(\mathbf{v}) - E(\mathbf{w}) - \langle \nabla E(\mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle| &\leq \int_0^1 | \langle L(\mathbf{w} + \tau(\mathbf{v} - \mathbf{w}) - \mathbf{w}), (\mathbf{v} - \mathbf{w}) \rangle | \, d\tau \\ &= L \int_0^1 \tau \| (\mathbf{v} - \mathbf{w}) \|^2 \, d\tau = \frac{L}{2} \| (\mathbf{v} - \mathbf{w}) \|^2 \end{aligned}$$

Suavidad & Cota para la Mejora

Aplicando la cota

$$|E(\mathbf{v}) - E(\mathbf{w}) - \nabla E(\mathbf{w})^T(\mathbf{v} - \mathbf{w})| \leq \frac{L}{2} \|\mathbf{w} - \mathbf{v}\|^2.$$

para $w = \mathbf{w}^{(t)}, v = \mathbf{w}^{(t+1)}$ obtenemos

$$E(\mathbf{w}^{(t+1)}) \leq E(\mathbf{w}^{(t)}) + \nabla E(\mathbf{w}^{(t)})^T(\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}) + \frac{L}{2} \|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\|^2$$

Recordando como tomamos los pasos ...

$$\begin{aligned} E(\mathbf{w}^{(t+1)}) &\leq E(\mathbf{w}^{(t)}) - \eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 + \frac{L\eta_t^2}{2} \|\nabla E(\mathbf{w}^{(t)})\|^2 \\ &= E(\mathbf{w}^{(t)}) - \eta_t \left(1 - \frac{L\eta_t}{2}\right) \|\nabla E(\mathbf{w}^{(t)})\|^2 \end{aligned}$$

O bien

$$\eta_t \left(1 - \frac{L\eta_t}{2}\right) \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)})$$



Suavidad & Cota para la Mejora

Si de alguna forma logramos que $\forall t \ 0 < \eta_t < 1/L$

$$\eta_t \left(1 - \frac{L\eta_t}{2} \right) > C > 0$$

De modo que:

$$\eta_t \left(1 - \frac{L\eta_t}{2} \right) \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)})$$

se puede transformar en

$$\|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \frac{E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)})}{C}$$

Convergencia

Sumando en t ...

$$\sum_{t=0}^T \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \frac{E(\mathbf{w}^{(0)}) - E(\mathbf{w}^{(T+1)})}{C} < \infty$$

Que implica:

$$\sum_{t=0}^{\infty} \|\nabla E(\mathbf{w}^{(t)})\|^2 = \lim_{T \rightarrow \infty} \sum_{t=0}^T \|\nabla E(\mathbf{w}^{(t)})\|^2 < \infty.$$

Que implica:

$$\lim_{t \rightarrow \infty} \|\nabla E(\mathbf{w}^{(t)})\|^2 = 0.$$

Notemos que:

$$\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\|^2 = \eta_t^2 \|\nabla E(\mathbf{w}^{(t)})\|^2$$



Convergencia de BP No Estocástico

Teorema

Si $E(\mathbf{w})$ es suave, no degenerada y η es suficientemente pequeña, el método anterior converge a un punto estacionario.

Punto Estacionario

$$\nabla_{\mathbf{w}} E = \frac{\delta E(\mathbf{w})}{\delta \mathbf{w}} = \mathbf{0}$$

Suavidad

$$\|\nabla E(\mathbf{w}) - \nabla E(\mathbf{v})\| \leq L \|\mathbf{w} - \mathbf{v}\|$$



Condición Sobre la Tasa

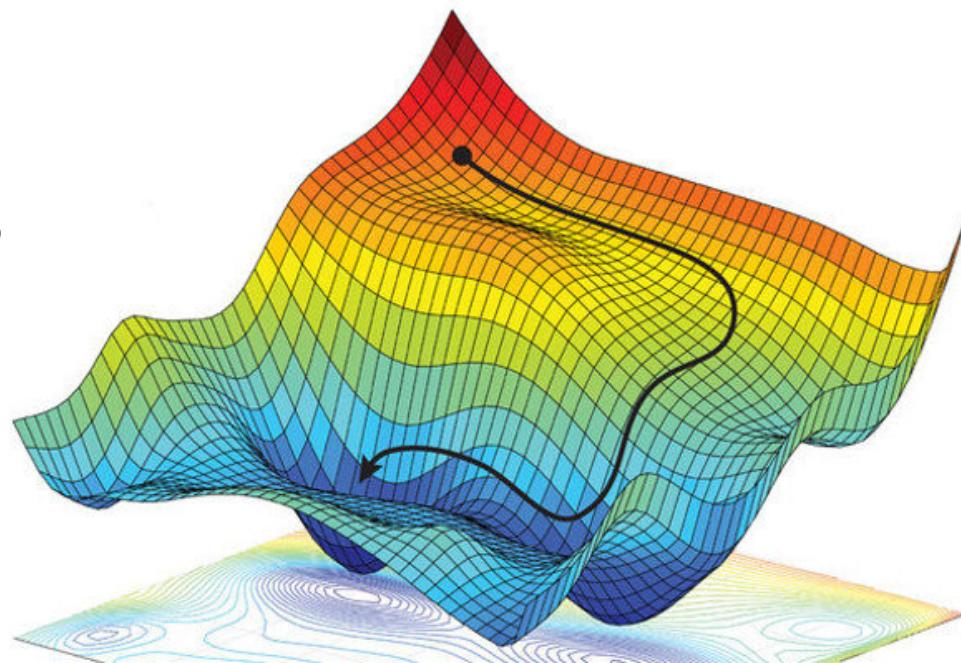
Si de alguna forma logramos que $\forall t \ 0 < \eta_t < 1/L$

$$\eta_t \left(1 - \frac{L\eta_t}{2} \right) > C > 0$$

L define la suavidad del gradiente:

$$\|\nabla E(\mathbf{w}) - \nabla E(\mathbf{v})\| \leq L\|\mathbf{w} - \mathbf{v}\|$$

Para funciones “complicadas” L no es constante en el dominio

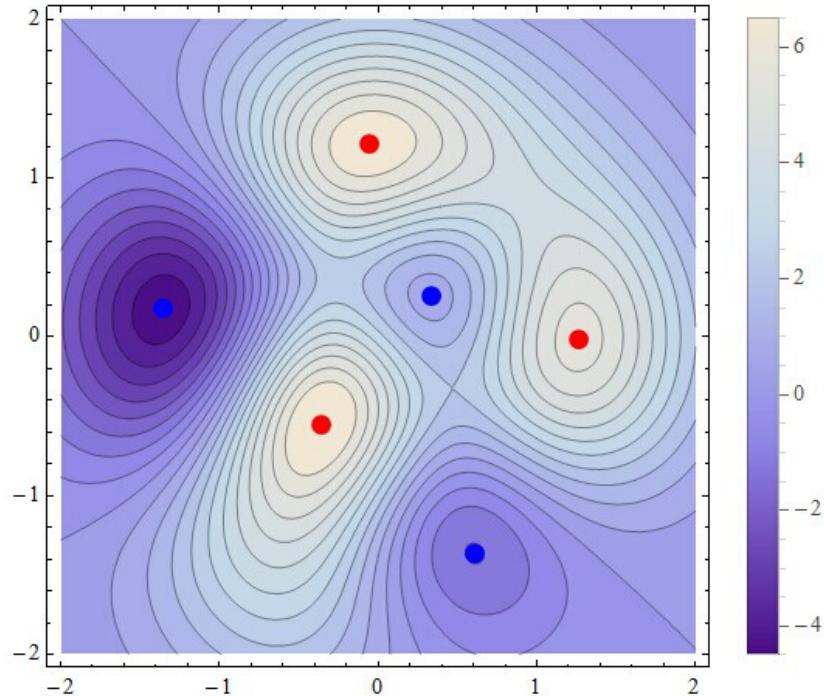


Condición Sobre la Tasa

La condición tasa grande/pequeña no es global. Si, como esperamos, a medida que transcurren las iteraciones nos acercamos a un buen mínimo local, la función debiese “suavizarse”.

Esto motiva un esquema de entrenamiento muy utilizado en la práctica:

- Se comienza con una tasa relativamente grande.
- Se reduce la tasa después de un número K de iteraciones multiplicando por una constante.



Decaimiento de la Tasa

- Una variante de este método reduce la tasa justamente cuando encuentra un *plateau* de la f.o. (no se mejora en P iteraciones).

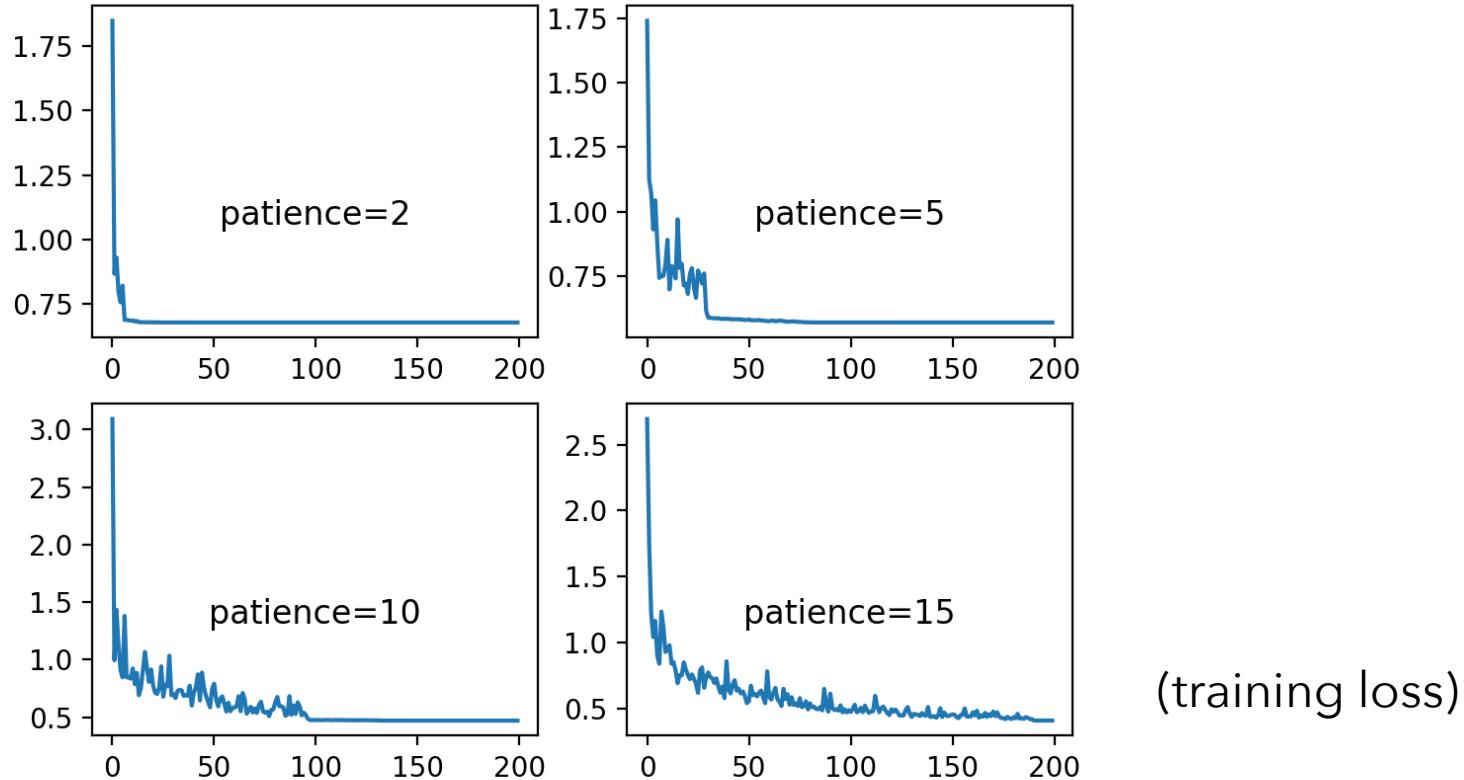
The screenshot shows the Keras documentation website. The left sidebar has a red 'K' logo and navigation links: About Keras, Getting started, Developer guides, Keras API reference (which is highlighted in black), Models API, Layers API, Callbacks API (which is highlighted in red), Data preprocessing, Optimizers, Metrics, Losses, Built-in small datasets, Keras Applications, and Utilities. The main content area has a search bar at the top. Below it, a breadcrumb trail shows '» Keras API reference / Callbacks API / ReduceLROnPlateau'. The title 'ReduceLROnPlateau' is displayed in large bold letters. Below the title is the heading 'ReduceLROnPlateau class'. A code block shows the Python code for the class:

```
tf.keras.callbacks.ReduceLROnPlateau(  
    monitor="val_loss",  
    factor=0.1,  
    patience=10,  
    verbose=0,  
    mode="auto",  
    min_delta=0.0001,  
    cooldown=0,  
    min_lr=0,  
    **kwargs  
)
```

Below the code, a description states: 'Reduce learning rate when a metric has stopped improving.' and 'Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced.'

Decaimiento de la Tasa

- Una variante de este método reduce la tasa justamente cuando encuentra un *plateau* de la f.o. (no se mejora en P iteraciones).



Progressive Decay

- Una forma popular de ordenar, “programar” y parametrizar este procedimiento se denomina “progressive decay”

$$\eta_t = \frac{\eta}{1 + \gamma t}$$

- El hiper-parámetro γ determina “cada cuanto” se decrece una cierta fracción, pero la regla se aplica de modo “suave” en el tiempo.
- Hoy se ven muchas variantes de este método clásico. Por ejemplo

$$\eta_t = \eta \cdot \alpha^{t/t_0}$$

Progressive Decay

Una forma popular de ordenar, “programar” y parametrizar este procedimiento se denomina “progressive decay”

The screenshot shows the Keras API reference page for the `InverseTimeDecay` class. The left sidebar has a red "Keras" logo and navigation links: About Keras, Getting started, Developer guides, Keras API reference (which is highlighted), Models API, Layers API, Callbacks API, Data preprocessing, Optimizers (which is highlighted with a red background), Metrics, Losses, Built-in small datasets, Keras Applications, Utilities, Code examples, and Why choose Keras?.

The main content area has a search bar at the top. Below it, the breadcrumb navigation shows: » Keras API reference / Optimizers / Learning rate schedules API / InverseTimeDecay. The title is **InverseTimeDecay**. The **InverseTimeDecay class** section contains the following code snippet:

```
tf.keras.optimizers.schedules.InverseTimeDecay(  
    initial_learning_rate, decay_steps, decay_rate, staircase=False, name=None  
)
```

A description follows: A `LearningRateSchedule` that uses an inverse time decay schedule. It explains that when training a model, it's often recommended to lower the learning rate as the training progresses. This schedule applies the inverse decay function to an optimizer step, given a provided initial learning rate. It requires a `step` value to compute the decayed learning rate. You can just pass a TensorFlow variable that you increment at each training step.

Another description follows: The schedule a 1-arg callable that produces a decayed learning rate when passed the current optimizer step. This can be useful for changing the learning rate value across different invocations of optimizer functions. It is computed as:

```
def decayed_learning_rate(step):  
    return initial_learning_rate / (1 + decay_rate * step / decay_step)
```



Progressive Decay

Una forma popular de ordenar, “programar” y parametrizar este procedimiento se denomina “progressive decay”

The screenshot shows the Keras API reference page for the `ExponentialDecay` class. The left sidebar has a red bar highlighting the `Optimizers` section. The main content area includes a search bar, a breadcrumb navigation bar, and the class documentation. It shows the class definition, a description of its purpose, and an example of how it's used.

Search Keras documentation...

» Keras API reference / Optimizers / Learning rate schedules API / ExponentialDecay

ExponentialDecay

`ExponentialDecay` class

```
tf.keras.optimizers.schedules.ExponentialDecay(  
    initial_learning_rate, decay_steps, decay_rate, staircase=False, name=None  
)
```

A LearningRateSchedule that uses an exponential decay schedule.

When training a model, it is often recommended to lower the learning rate as the training progresses. This schedule applies an exponential decay function to an optimizer step, given a provided initial learning rate.

The schedule a 1-arg callable that produces a decayed learning rate when passed the current optimizer step. This can be useful for changing the learning rate value across different invocations of optimizer functions. It is computed as:

```
def decayed_learning_rate(step):  
    return initial_learning_rate * decay_rate ^ (step / decay_steps)
```



Convergencia de BP Estocástico

En este caso, utilizamos una versión aproximada del gradiente de la f.o.

- 1 Elegir un valor para $\mathbf{w}^{(0)}$;
- 2 **for** $t = 1, \dots, T$ **do**
- 3 | $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta_t \tilde{\nabla} E(\mathbf{w}^{(t)})$;
- 4 **end**

$$\tilde{\nabla} E(\mathbf{w}) = \nabla L(f(\mathbf{x}^{(i)}), y^{(i)})$$

gradiente aproximado

$$\nabla E(\mathbf{w}) = \sum_i \nabla L(f(\mathbf{x}^{(i)}), y^{(i)})$$

gradiente verdadero



Convergencia de BP Estocástico

Sin embargo, el gradiente de la f.o. es ya una aproximación del gradiente ideal

$$\tilde{\nabla}E(\mathbf{w}) = \nabla L(f(\mathbf{x}^{(i)}), y^{(i)})$$

gradiente aproximado

$$\nabla E(\mathbf{w}) = \sum_i \nabla L(f(\mathbf{x}^{(i)}), y^{(i)})$$

gradiente verdadero

$$\begin{aligned}\nabla R(\mathbf{w}) &= \nabla \int L(f(\mathbf{x}), y) dP(\mathbf{x}, y) \\ &= \int \nabla L(f(\mathbf{x}), y) dP(\mathbf{x}, y)\end{aligned}$$

gradiente ideal



Convergencia de BP Estocástico

Notemos que:

$$\begin{aligned}\mathbb{E} [\tilde{\nabla} E(\mathbf{w})] &= \mathbb{E} [\nabla L(f(\mathbf{x}^{(i)}), y^{(i)})] \\ &= \int \nabla L(f(\mathbf{x}), y) dP(x, y) = \nabla R(\mathbf{w})\end{aligned}$$

Esto sugiere que es posible demostrar algún tipo de convergencia "en valor esperado". En efecto, demostraremos que:

$$\min_{t=1, \dots, T} \mathbb{E} \left(\|\nabla E(\mathbf{w}^{(t)})\|^2 \right) \xrightarrow{T \rightarrow \infty} 0$$



Convergencia de BP Estocástico

Haremos esto asumiendo que, además de nuestra observación anterior $\mathbb{E}(\tilde{\nabla}E) = \nabla E$ se verifican algunas condiciones sobre la f.o, la forma en que aproximamos el gradiante y la tasa:

Suavidad

$$\|\nabla E(\mathbf{w}) - \nabla E(\mathbf{v})\| \leq L\|\mathbf{w} - \mathbf{v}\|$$

Estabilidad

$$\mathbb{E} \left(\|\tilde{\nabla}E(\mathbf{w}^{(t)})\|^2 \mid \mathbf{w}^{(t)} \right) \leq \sigma^2 < \infty$$

Condiciones de Robbins-Monro

$$\sum_t \eta_t = \infty, \quad \sum_t \eta_t^2 < \infty$$



Convergencia de BP Estocástico

Al igual que antes, partimos explotando la suavidad de la f.o.

$$\begin{aligned} E(\mathbf{w}^{(t+1)}) &\leq E(\mathbf{w}^{(t)}) + \nabla E(\mathbf{w}^{(t)})^T (\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}) + \frac{L}{2} \|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\|^2 \\ &= E(\mathbf{w}^{(t)}) - \eta_t \nabla E(\mathbf{w}^{(t)})^T \tilde{\nabla} E(\mathbf{w}^{(t)}) + \frac{L\eta_t^2}{2} \|\tilde{\nabla} E(\mathbf{w}^{(t)})\|^2 \end{aligned}$$

Notando ahora que y usando el supuesto de estabilidad del gradiente

$$\mathbb{E} \left(\nabla E(\mathbf{w}^{(t)})^T \tilde{\nabla} E(\mathbf{w}^{(t)}) \mid \mathbf{w}^{(t)} \right) = \|\nabla E(\mathbf{w}^{(t)})\|^2 \quad \mathbb{E} \left(\|\tilde{\nabla} E(\mathbf{w}^{(t)})\|^2 \mid \mathbf{w}^{(t)} \right) \leq \sigma^2 < \infty$$

Obtenemos ...



Convergencia de BP Estocástico

Al igual que antes, partimos explotando la suavidad de la f.o.

$$\begin{aligned} E(\mathbf{w}^{(t+1)}) &\leq E(\mathbf{w}^{(t)}) + \nabla E(\mathbf{w}^{(t)})^T (\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}) + \frac{L}{2} \|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\|^2 \\ &= E(\mathbf{w}^{(t)}) - \eta_t \nabla E(\mathbf{w}^{(t)})^T \tilde{\nabla} E(\mathbf{w}^{(t)}) + \frac{L\eta_t^2}{2} \|\tilde{\nabla} E(\mathbf{w}^{(t)})\|^2 \end{aligned}$$

Notando ahora que y usando el supuesto de estabilidad del gradiente

$$\mathbb{E} \left(\nabla E(\mathbf{w}^{(t)})^T \tilde{\nabla} E(\mathbf{w}^{(t)}) \mid \mathbf{w}^{(t)} \right) = \|\nabla E(\mathbf{w}^{(t)})\|^2 \quad \mathbb{E} \left(\|\tilde{\nabla} E(\mathbf{w}^{(t)})\|^2 \mid \mathbf{w}^{(t)} \right) \leq \sigma^2 < \infty$$

Obtenemos ...

$$\eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \mathbb{E} \left(E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)}) \mid \mathbf{w}^{(t)} \right) + \frac{\sigma^2 L \eta_t^2}{2}$$

Convergencia de BP Estocástico

Desde acá

$$\eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \mathbb{E} \left(E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)}) \middle| \mathbf{w}^{(t)} \right) + \frac{\sigma^2 L \eta_t^2}{2}$$

Basta tomar valor esperado incondicional y aprovechar la regla del valor esperado total

$$\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|Y))$$

Para obtener

$$\sum_{t=1}^T \eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \mathbb{E} \left(E(\mathbf{w}^{(0)}) - E(\mathbf{w}^{(T+1)}) \right) + \frac{\sigma^2 L \sum_{t=1}^T \eta_t^2}{2}$$



Convergencia de BP Estocástico

Desde acá

$$\eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \mathbb{E} \left(E(\mathbf{w}^{(t)}) - E(\mathbf{w}^{(t+1)}) \middle| \mathbf{w}^{(t)} \right) + \frac{\sigma^2 L \eta_t^2}{2}$$

Basta tomar valor esperado incondicional y aprovechar la regla del valor esperado total

$$\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|Y))$$

Para obtener

$$\sum_{t=1}^T \eta_t \|\nabla E(\mathbf{w}^{(t)})\|^2 \leq \mathbb{E} \left(E(\mathbf{w}^{(0)}) - E(\mathbf{w}^{(T+1)}) \right) + \frac{\sigma^2 L \sum_{t=1}^T \eta_t^2}{2}$$



Convergencia de BP Estocástico

Desde acá aplicamos cotas sencillas

$$\begin{aligned} \sum_{t=1}^T \eta_t \mathbb{E} \left(\|\nabla E(\mathbf{w}^{(t)})\|^2 \right) &\leq \mathbb{E} \left(E(\mathbf{w}^{(0)}) - E(\mathbf{w}^{(T+1)}) \right) + \frac{\sigma^2 L \sum_{t=1}^T \eta_t^2}{2} \\ &\leq \left(E(\mathbf{w}^{(0)}) - E(\mathbf{w}^*) \right) + \frac{\sigma^2 L \sum_{t=1}^T \eta_t^2}{2} \\ &= \frac{\Delta + \sigma^2 L \sum_{t=1}^T \eta_t^2}{2} \\ \Rightarrow \min_{t=1,\dots,T} \mathbb{E} \left(\|\nabla E(\mathbf{w}^{(t)})\|^2 \right) &\leq \frac{\Delta + \sigma^2 L \sum_{t=1}^T \eta_t^2}{2 \sum_{t=1}^T \eta_t} \end{aligned}$$

Usando las condiciones de Robbins Monroe, obtenemos la convergencia

Condiciones de Robbins-Monro

$$\min_{t=1,\dots,T} \mathbb{E} \left(\|\nabla E(\mathbf{w}^{(t)})\|^2 \right) \xrightarrow{T \rightarrow \infty} 0$$

$$\sum_t \eta_t = \infty, \quad \sum_t \eta_t^2 < \infty$$



Condiciones de Convergencia

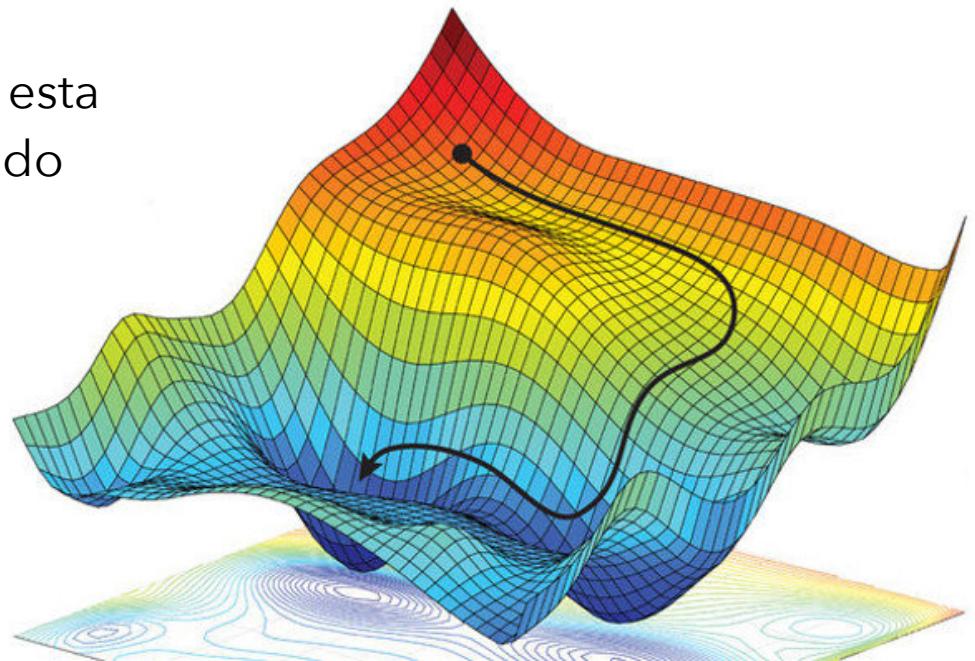
Obtenemos nuevamente que la convergencia depende de condiciones sobre la forma de la f.o y de la estabilidad de los gradientes que calculamos

Para funciones “complicadas” tanto L como σ^2 pueden cambiar durante la optimización, lo que sugiere el uso de mecanismos que estén atentos tanto a la magnitud del gradiente como a su segundo momento.

La condición sobre la tasa significa que esta debe decrecer pero no demasiado rápido

Condiciones de Robbins-Monro

$$\sum_t \eta_t = \infty, \quad \sum_t \eta_t^2 < \infty$$



Condiciones sobre la Tasa

Notemos que si η_t decrece como $\mathcal{O}(1/t)$ las condiciones de Robbins Monroe se satisfacen

$$\sum_t \eta_t = \infty \quad \sum_t \eta_t^2 < \infty$$

Lo que motiva progressive decay: $\eta_t = \frac{\eta}{1 + \gamma t}$

Ocurre esto para las variantes de esta regla?

$$\eta_t = \eta \cdot \alpha^{t/t_0}$$



Entonces ...

- Dadas ciertas condiciones de suavidad de la f.o. y de buen comportamiento de la tasa de aprendizaje, es “fácil” probar la convergencia de BP a un punto estacionario que, esperamos, sea un mínimo local.
- Una forma popular de calibrar la tasa es diseñar un programa (schedule) que reduzca gradualmente su valor a medida que progresa el aprendizaje. Una versión clásica de esta regla es progressive decay.
- Dadas ciertas condiciones de suavidad de la f.o., decaimiento de la tasa de aprendizaje, y estabilidad de nuestras aproximaciones al gradiente, es “fácil” probar la convergencia de BP estocástico a un punto estacionario “en valor esperado” y en algún momento del entrenamiento.
- Dado que la forma loca de la f.o. puede cambiar mucho durante el entrenamiento es sensato pensar en reglas adaptativas de calibración de la tasa que se informen sobre el primer y segundo momento del gradiente.

