¿Por qué la solución al problema de optimalidad funciona?

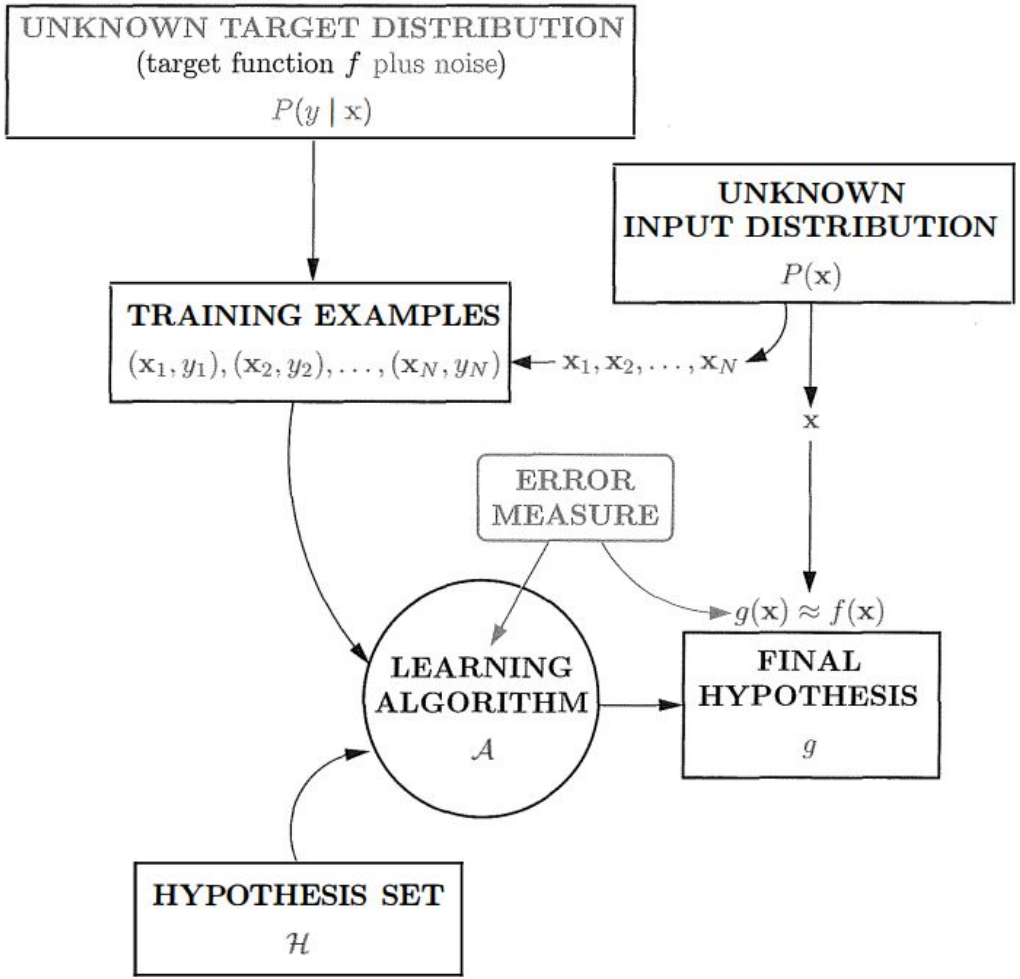
… un poco de contexto…

Contenido

1. Contexto
    a. Función de pérdida y riesgo empírico
    b. Gradiente en descenso estocástico
    c. Convexidad
2. Problemas
    a. Ejemplo
3. Posibles explicaciones
    a. Analogía física
    b. Caminos y conjuntos de nivel
    c. Convergencia

Recordemos

Para cuantificar qué tanto se aproxima una hipótesis g a la función objetivo f, usamos una **medida del error**.



Recordemos

La **función de pérdida** penaliza la mala clasificación o aproximación de un ejemplo particular z, por una función f_z en el espacio de hipótesis. La podemos notar así:

$$\mathcal{L}(f_s, z)$$

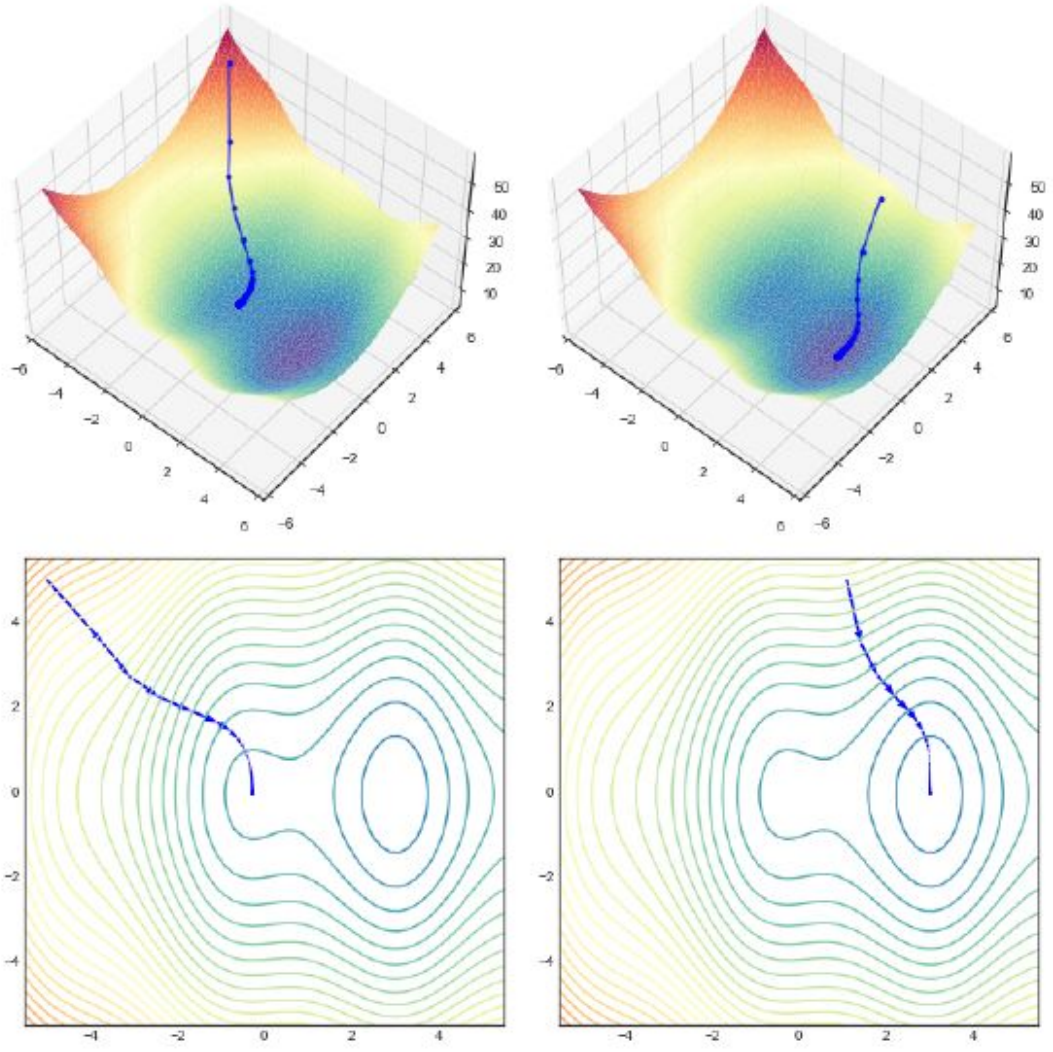
En el proceso de aprendizaje, intentamos minimizar una **función de costo**. Un ejemplo particular de una función de costo es el **riesgo empírico**, definido como

$$\widehat{\mathcal{R}}_s(f) := \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(f, z^{(i)}).$$

Recordemos

Gradiente en descenso:

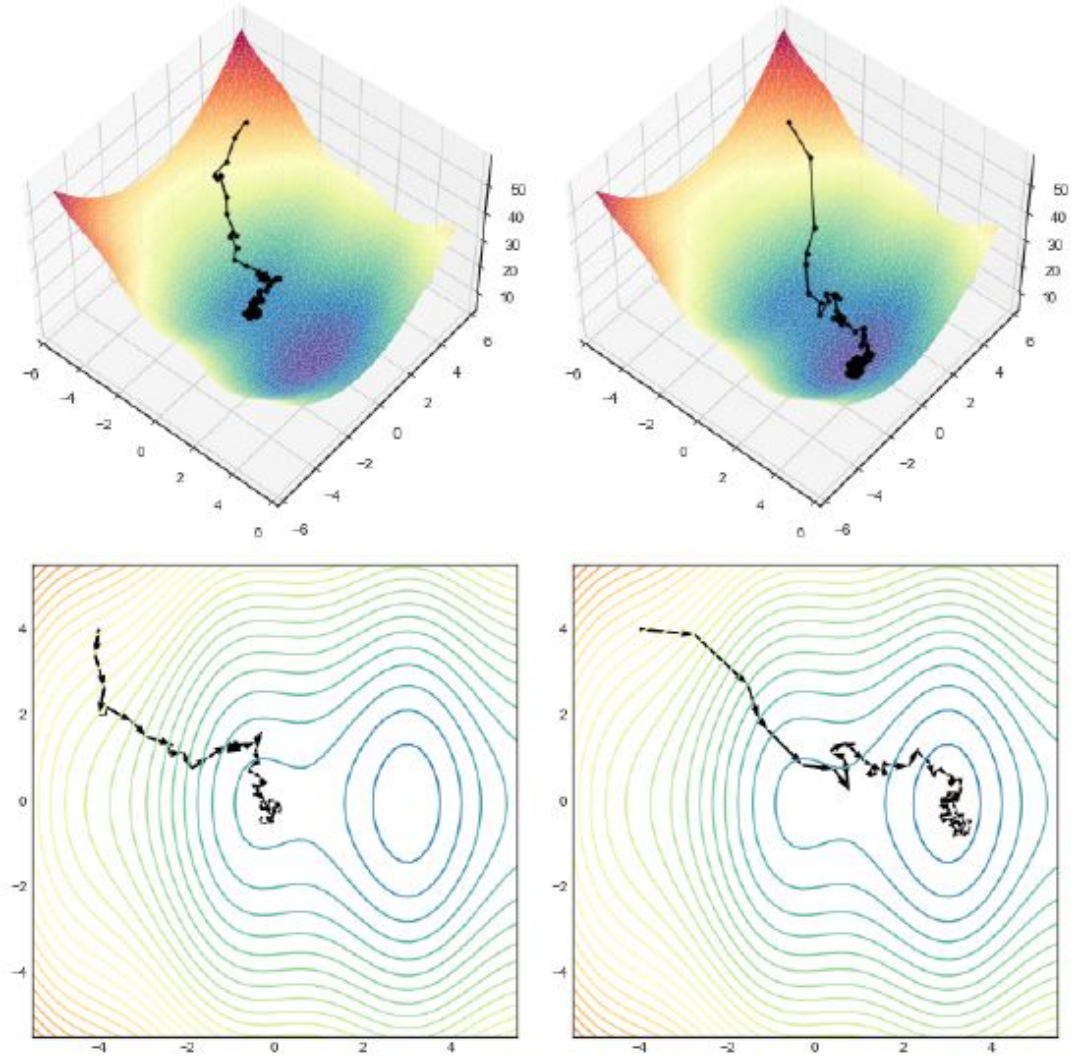
```
while True:
    theta_grad = evaluate_gradient(J,corpus,theta)
    theta = theta - alpha * theta_grad
```



Recordemos

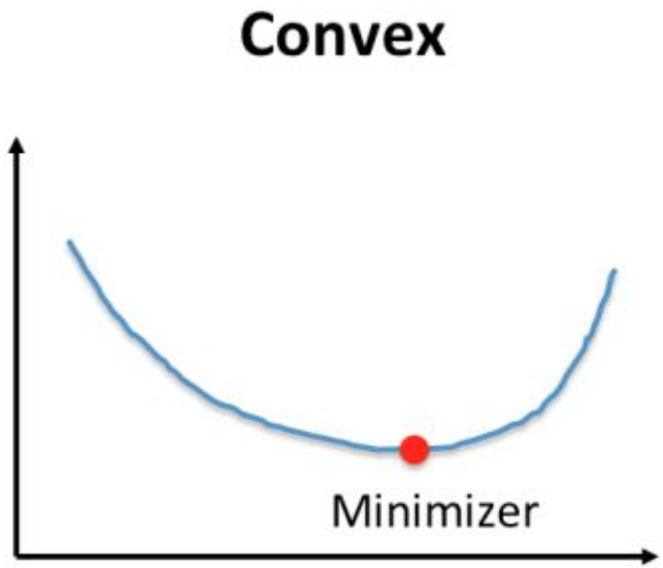
Gradiente en descenso estocástico:

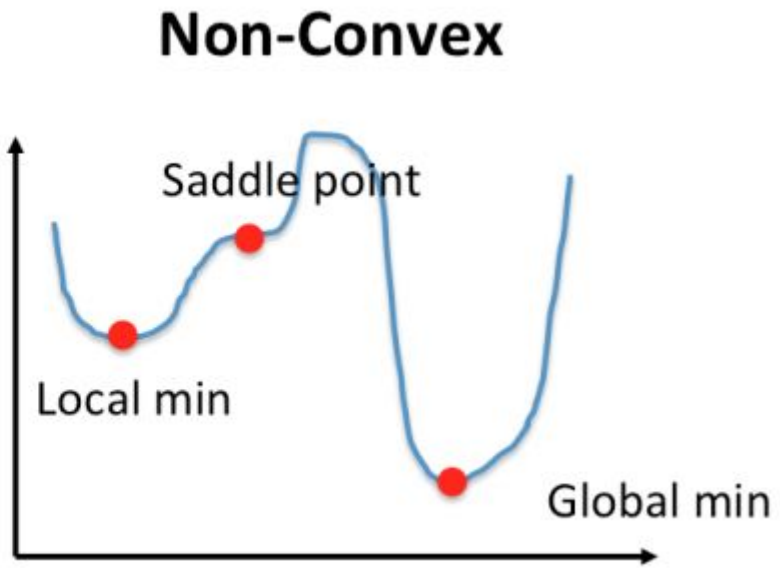
```
while True:
    window = sample_window(corpus)
    theta_grad = evaluate_gradient(J,window,theta)
    theta = theta - alpha * theta_grad
```



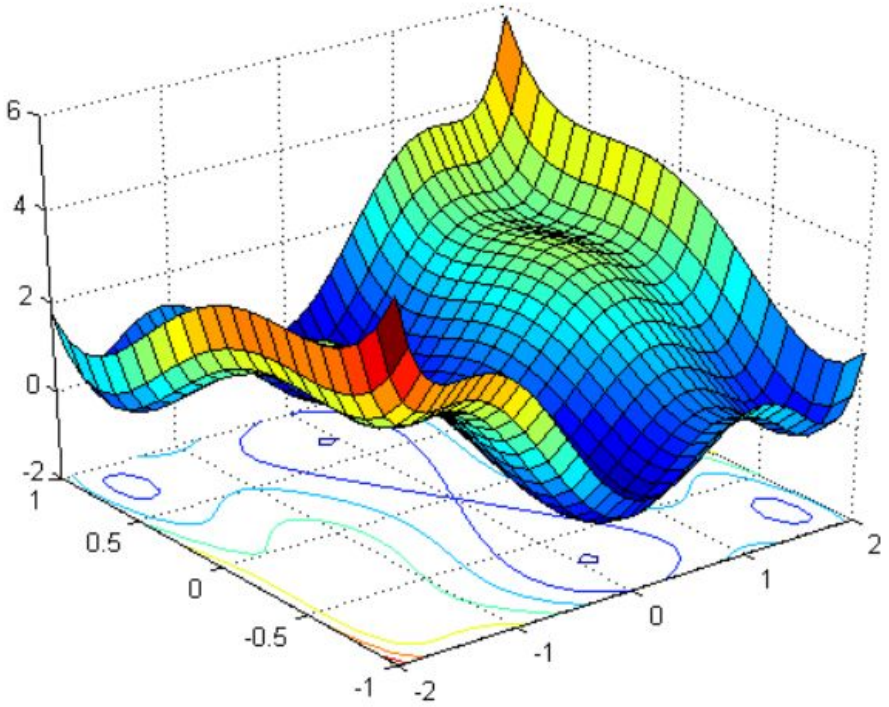
Problemas convexos

Para las funciones convexas, todo mínimo local es mínimo global.

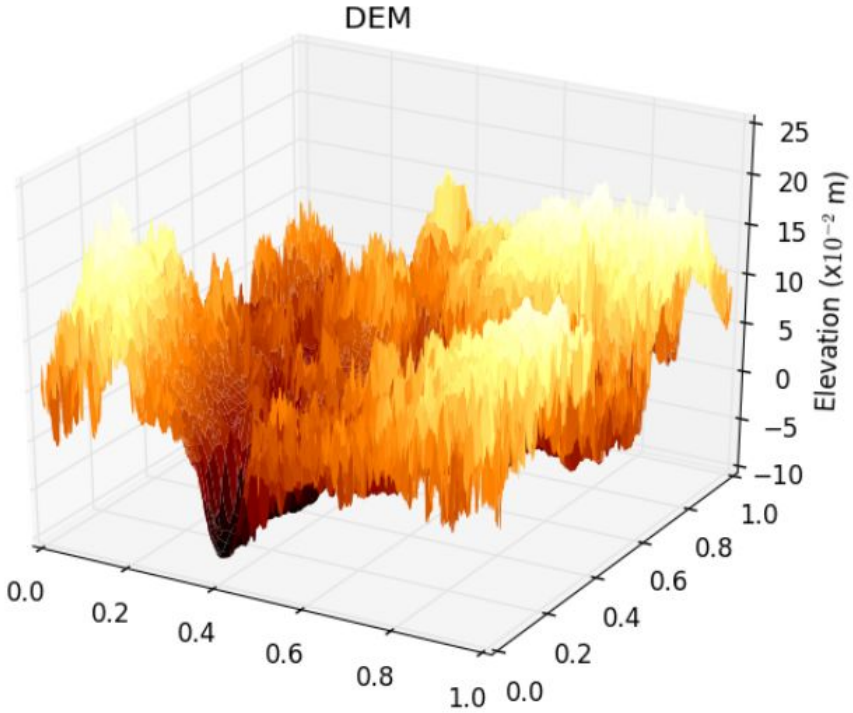




En general, las **funciones de costo** (riesgo empírico) de las redes neuronales **NO son convexas.**



Función de pérdida en redes neuronales



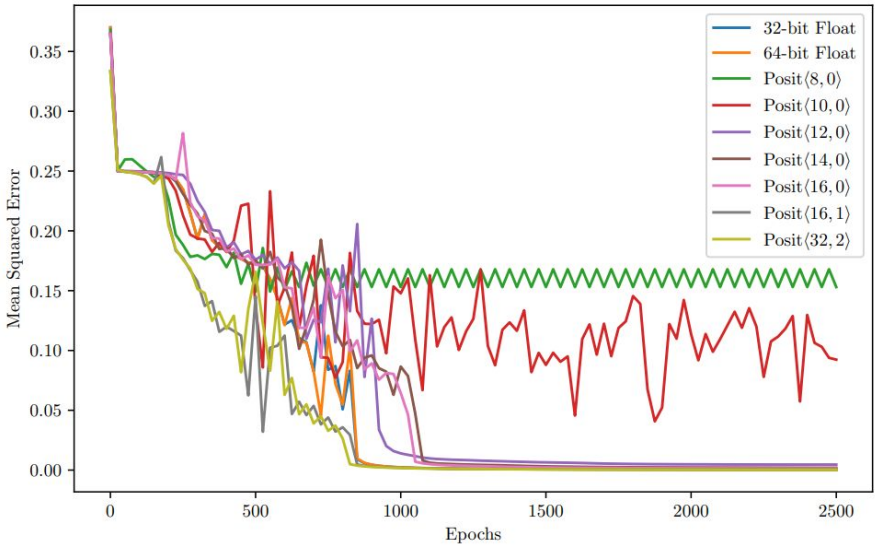


Figure 5: Loss function along the NN training.

Ejemplo 1.20

Consideramos la red neuronal

$$\Phi(x,\theta) = \theta_1 \varrho_R(\theta_3 x + \theta_5) + \theta_2 \varrho_R(\theta_4 x + \theta_6), \quad \theta \in \mathbb{R}^6, x \in \mathbb{R}$$

con función de activación ReLu:

$$\varrho_R(x) = \max\{0, x\}$$

Para los parámetros $\theta = (1,-1,1,1,1,0), \quad \bar{\theta} = (-1,1,1,1,0,1)$

la red funciona de la misma manera.

$$\Phi(x,\theta) = \Phi(x,\bar{\theta})$$

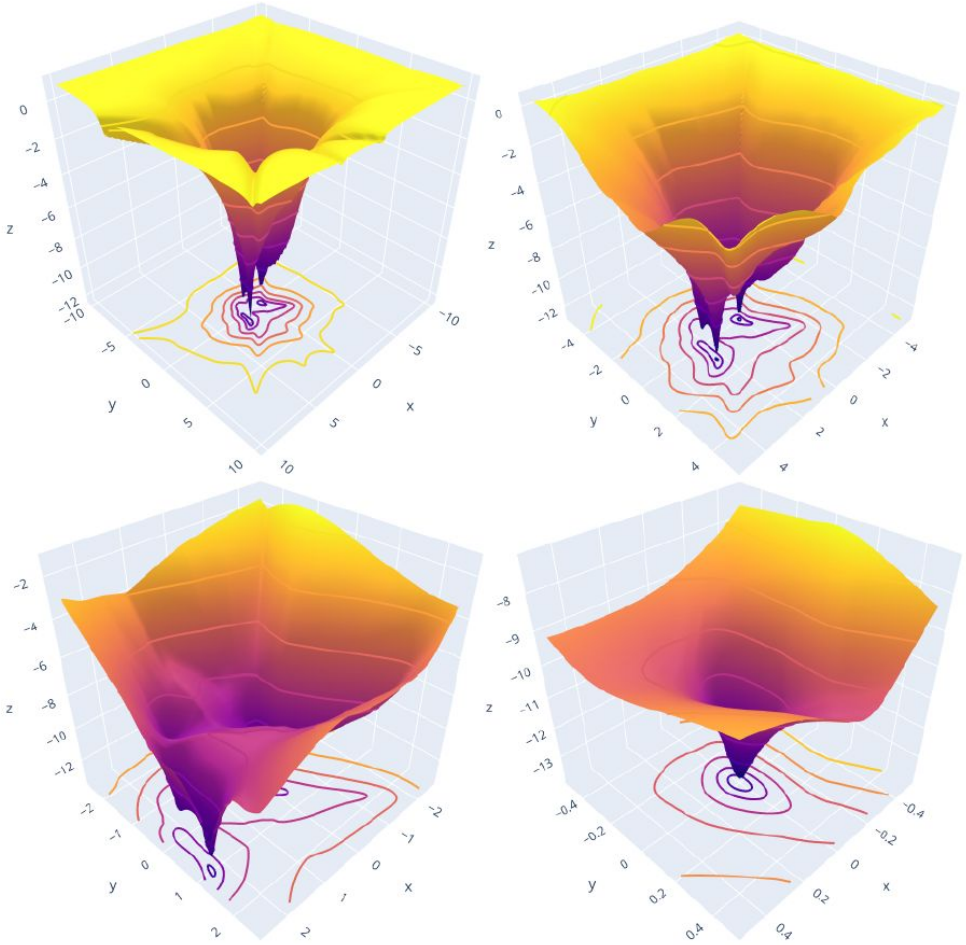


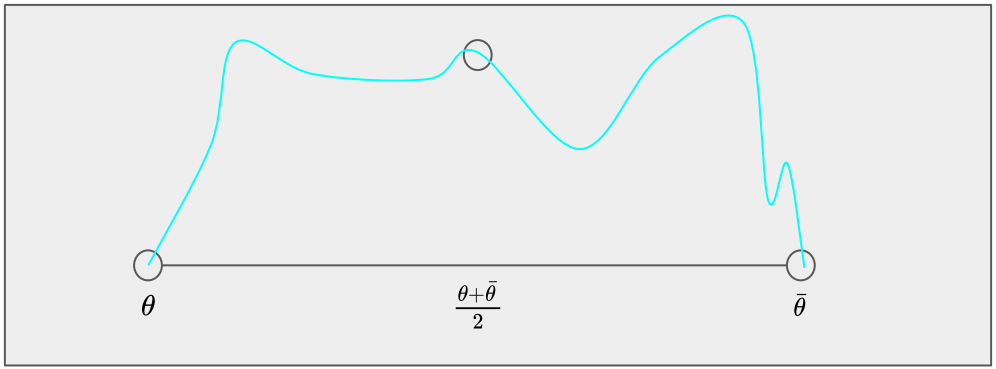
Figure 1.5: Two-dimensional projection of the loss landscape of a neural network with four layers and ReLU activation function on four different scales. From top-left to bottom-right, we zoom into the global minimum of the landscape.

Se presentan problemas

En particular este es no convexo pues: $s = ((-1, 0), (1, 1))$

$$\hat{\mathcal{R}}_s(\Phi(\cdot, \theta)) = \hat{\mathcal{R}}_s(\Phi(\cdot, \bar{\theta})) = 0,$$

$\frac{\theta+\bar{\theta}}{2} = (0, 0, 1, 1, 1/2, 1/2)$ $\Phi(\cdot, \frac{\theta+\bar{\theta}}{2}) = 0$ $\hat{\mathcal{R}}_s(\Phi(\cdot, \frac{\theta+\bar{\theta}}{2})) = \frac{1}{2}$



Más problemas

1. Varios mínimos locales.
2. Puntos de silla (algunos de alto orden, se desvanece la Hessiana).
3. Gradiente muy pequeño -> Toma mucho tiempo.
4. En general para las L^p normas el conjunto de redes neuronales está lejos de ser convexo y cerrado.

A pesar de todos estos problemas, el aprendizaje profundo usando SGD funciona muy bien en la práctica. Este fenómeno no puede ser explicado bajo el marco clásico de la teoría del aprendizaje.

Algunas posibles explicaciones

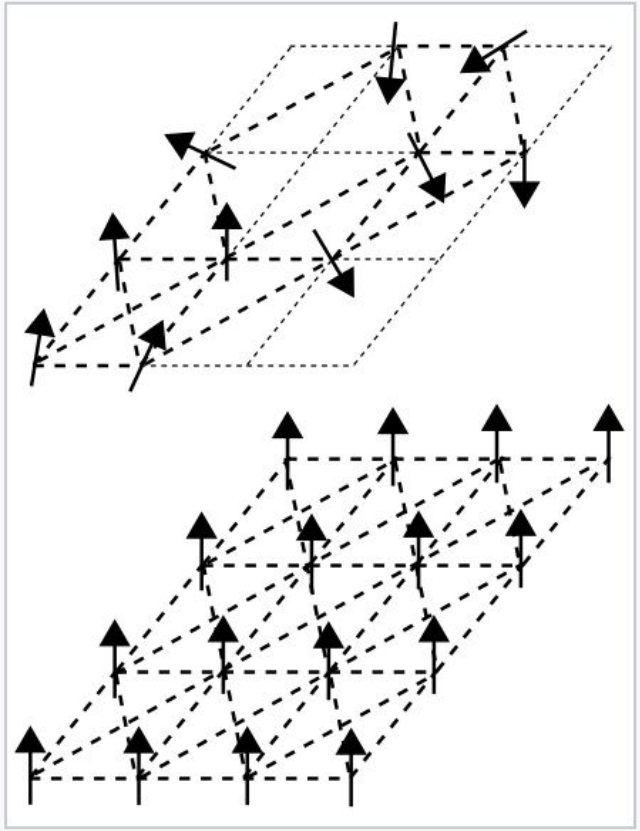
1. **Análisis del “paisaje” (gráfica) de la función de pérdida.**

2. **Convergencia de SGD para redes neuronales sobre-parametrizadas *(Demostrable)*.**

1.1 Interpretación del vidrio de espín.

El **hamiltoniano** de un sistema es un operador (función) que corresponde a la energía total del sistema (energía potencial + energía cinética).

El **vidrio de espín** es un sistema magnético caracterizado por la aleatoriedad de los espíns, entre otras cosas.



Schematic representation of the **random** spin structure of a **spin glass** (top) and the **ordered** one of a **ferromagnet** (bottom)

Para modelos de **vidrio de espín,** el **hamiltoniano** tiene cierta propiedad sobre la matriz Hessiana cuando esta es evaluada en sus puntos críticos.

Esta propiedad implica una **relación** entre la **inestabilidad** de un punto crítico y su **diferencia respecto al mínimo global**. Si usamos SGD y en alguna de sus iteraciones nos encontramos con un mínimo local de este hamiltoniano, es probable que su valor sea cercano al del mínimo global.

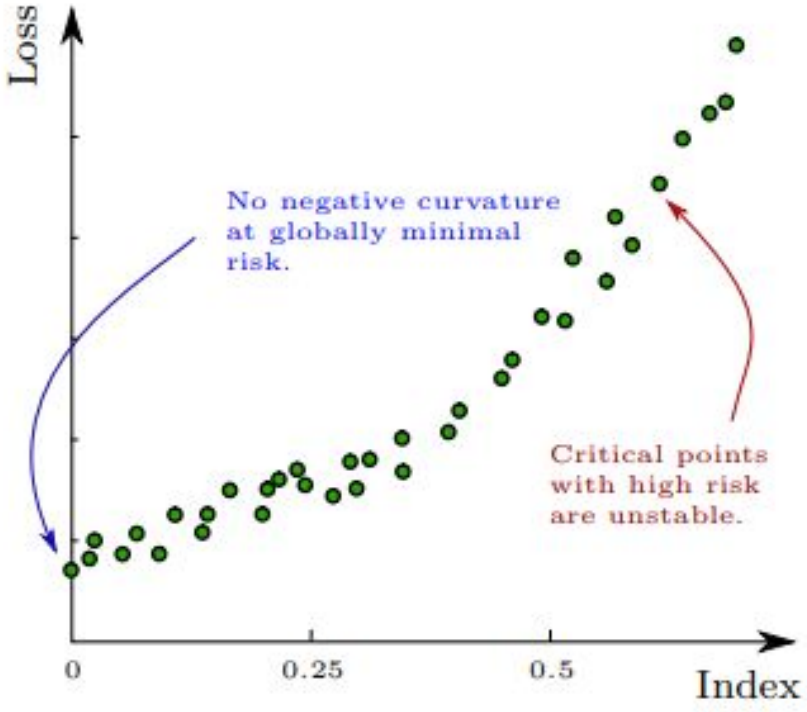
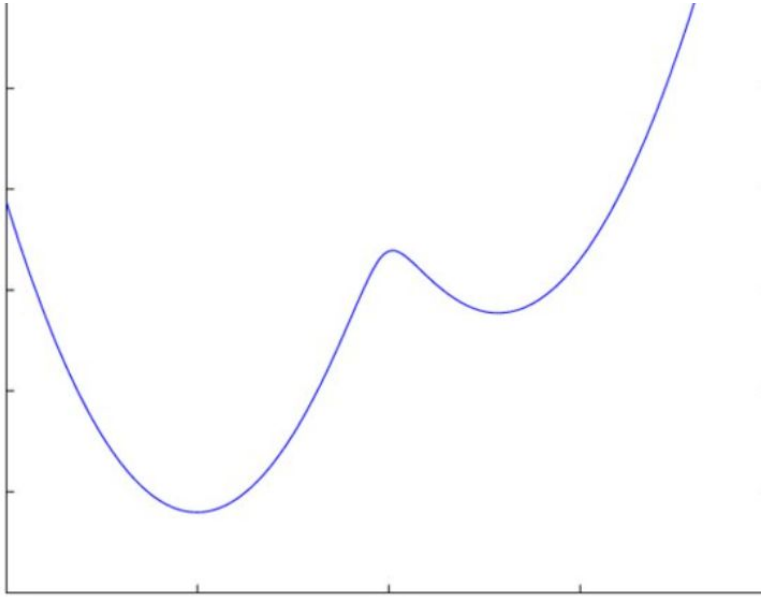


Figure 5.1: Sketch of the distribution of critical points of the Hamiltonian of a spin glass model.

Punto crítico estable



Punto crítico no estable

¿Cómo se relaciona lo anterior con la función de pérdida o riesgo de las redes neuronales?

Asumiendo algunas simplificaciones, se demostró que **la función de pérdida de una red neuronal con inputs aleatorios** (parámetros aleatorios) puede ser considerado como el hamiltoniano de un modelo de vidrio de espín.

Algunas de las suposiciones realizadas para concluir esta relación no siempre se mantienen en la práctica, pero las propiedades mencionadas respecto a los puntos críticos sí se presentan usualmente en la práctica.

1.2 Caminos a lo largo del espacio de parámetros

Otra línea centra sus esfuerzos en estudiar caminos a lo largo del espacio de parámetros. En particular, los caminos de interés son aquellos en los que el riesgo empírico asociado a los parámetros del camino, es monótono. La existencia de un camino en el que el riesgo empírico es no-creciente que vaya de cualquier punto al mínimo global, garantizaría la no existencia de mínimos no globales.

En primer lugar, debemos recordar que lo que queremos es minimizar el riesgo empírico

$$F(\theta) = \frac{1}{L} \sum_{i=1}^{L} \|\Phi(x_i; \theta) - y_i\|^2 + \kappa \mathcal{R}(\theta),$$

donde L es el tamaño del conjunto de entrenamiento, θ contiene los pesos y sesgos de todas las capas de la red, $\mathcal{R}(\theta)$ es un término de regularización, y $\Phi(x; \theta)$ encapsula la representación de características que usa el vector de parámetros θ.

Luego podemos pasar a definir uno de los elementos más importantes del análisis que se va a realizar; los conjuntos de nivel:

$$\Omega_F(\lambda) = \{\theta \in \mathbb{R}^S : F(\theta) \leq \lambda\}.$$

Proposición 1 *si $\Omega_F(\lambda)$ es conexo para todo λ entonces todo mínimo local de F es un mínimo global.*

Esta afirmación nos brinda una condición suficiente para prevenir la existencia de mínimos no globales, sin embargo, no es una condición necesaria ya que se podrían tener mínimos locales aislados en el mismo nivel de energía, como lo es el caso de las redes multicapa.

Ahora bien, queremos estudiar cuáles modelos tienen conjuntos de nivel conexos. El primer resultado dice que para una red multicapa de la forma

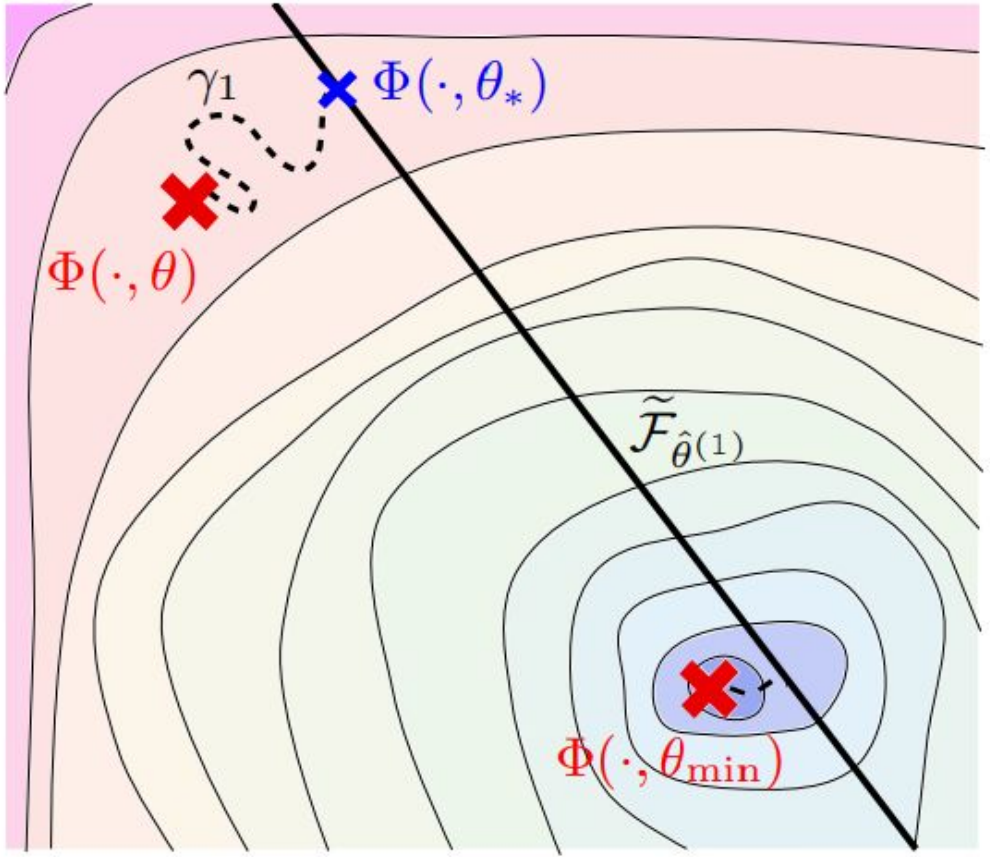
$$\Phi(x;\theta) = W_K \dots W_1 x,$$

donde el término de regularización es $\mathcal{R}(\theta) = \|\theta\|^2$, cada mínimo local es un mínimo global

pero…
para redes del tipo

$$\Phi(x;\theta) = W_K \rho W_{k-1} \rho \dots \rho W_1 x,$$

donde $\rho(z) = \text{máx}(0, z)$ es una no linealidad RELU. Se pueden construir contraejemplos que muestran que modelos no lineales como el anterior no tienen la propiedad de conexidad global, y la construcción de estos contraejemplos nos muestra que el problema optimización depende de la distribución de los datos [5].


γ_1
$\Phi(\cdot, \theta_*)$
$\Phi(\cdot, \theta)$
$\widetilde{\mathcal{F}}_{\hat{\theta}^{(1)}}$
$\Phi(\cdot, \theta_{\min})$

Resultado principal

El resultado considera un escenario asintótico dado por una neurona fija m de la capa de la red. Dados dos parámetros $\theta^A = (W_1^A, W_2^A) \in \mathcal{W}$ y $\theta^B = (W_1^B, W_2^B)$, con $F(\theta^{\{A,B\}}) \leq \lambda$, se muestra que existe un camino continuo $\gamma : [0,1] \to \mathcal{W}$ que conecta a θ^A con θ^B, tal que el riesgo está acotado uniformemente por $\max(\lambda, \epsilon)$, donde ϵ decrece de acuerdo a la sobreparametrización del modelo.

Del resultado anterior se puede obtener que a medida que m incrementa, el gap ϵ satisface que $\epsilon = \mathcal{O}(m^{-1/n})$, y por lo tanto los conjuntos de nivel se vuelven conexos en todos los niveles de energía.

Resultados

Abstract

We show that for a single neuron with the logistic function as the transfer function the number of local minima of the error function based on the square loss can grow exponentially in the dimension.

Abstract

We consider the optimization problem associated with training simple ReLU neural networks of the form $\mathbf{x} \mapsto \sum_{i=1}^{k} \max\{0, \mathbf{w}_i^\top \mathbf{x}\}$ with respect to the squared loss. We provide a computer-assisted proof that even if the input distribution is standard Gaussian, even if the dimension is arbitrarily large, and even if the target values are generated by such a network, with orthonormal parameter vectors, the problem can still have spurious local minima once $6 \leq k \leq 20$. By a concentration of measure argument, this implies that in high input dimensions, *nearly all* target networks of the relevant sizes lead to spurious local minima. Moreover, we conduct experiments which show that the probability of hitting such local minima is quite high, and increasing with the network size. On the positive side, mild over-parameterization appears to drastically reduce such local minima, indicating that an over-parameterization assumption is necessary to get a positive result in this setting.

very non-convex and non-closed set. Also, the map $\theta \mapsto \Phi_a(\cdot, \theta)$ is not a quotient map, i.e., not continuously invertible when accounting for its non-injectivity. In addition, in various situations finding the global optimum of the minimization problem is shown to be NP-hard in general [BR89, Jud90, Ším02]. In Figure 1.5 we show the two-dimensional projection of a loss landscape, i.e., the projection of the graph of the function $\theta \mapsto \widehat{\mathcal{R}}_s(\Phi(\cdot, \theta))$. It is apparent from the visualization that the problem exhibits more than one minimum. We also want to add that in practice one neglects that the loss is only almost everywhere differentiable in case of piecewise smooth activation functions, such as the ReLU, although one could resort to subgradient methods [KL18].

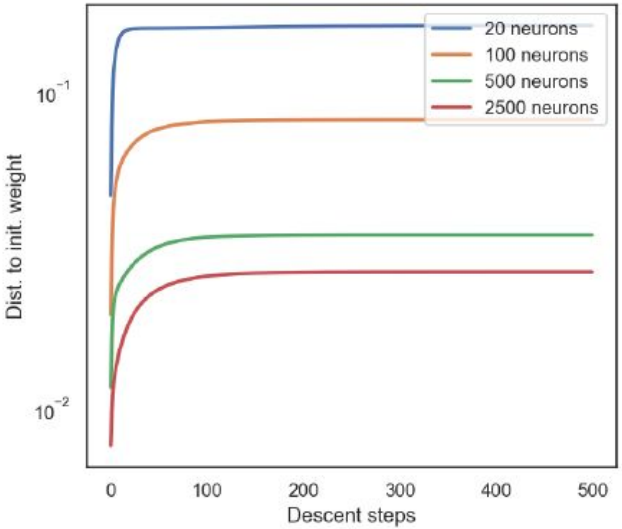
In view of these considerations, the classical framework presented in Subsection 1.2.1 offers no explanation as to why deep learning works in practice. Indeed, in the survey [OM98, Section 1.4] the state of the art in 1998 was summarized by the following assessment: "There is no formula to guarantee that (1) the NN will converge to a good solution, (2) convergence is swift, or (3) convergence even occurs at all."

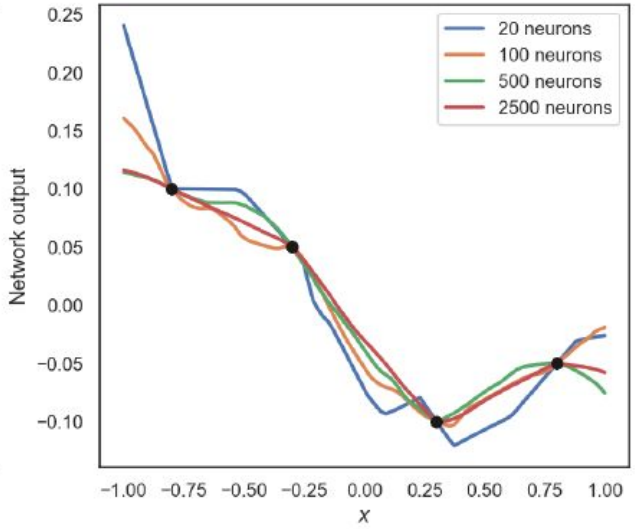
Peeeero

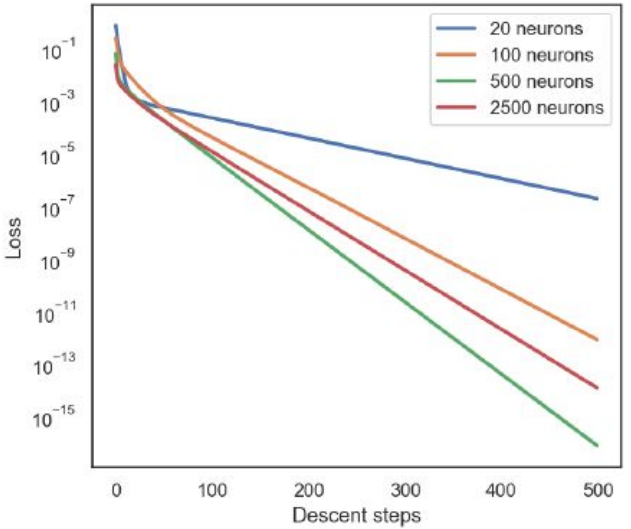
Nonetheless, in applications, not only would an explanation of when and why SGD converges be extremely desirable, convergence is also quite often observed even though there is little theoretical explanation for it in the classical set-up.

2. Convergencia de SGD para redes neuronales sobre-parametrizadas

¿Qué se ha observado desde el punto de vista de la parametrización de las NNs?







Un modelo simple

Los efectos de cambio en la sobre-parametrización son estudiados a través del mismo modelo de NN de dos capas sin sesgo presentada en la sección 1.2:

$$\mathbb{R}^d \ni x \mapsto \Phi(x,\theta) := \sum_{j=1}^{n} \theta_j^{(2)} \varrho\big(\langle \theta_j^{(1)}, \begin{bmatrix} x \\ 1 \end{bmatrix} \rangle\big),$$

Para la función de pérdida cuadrática y conjunto de datos s definimos el riesgo empírico como:

$$\boxed{r(\theta)} = \widehat{\mathcal{R}}_s(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\Phi(x^{(i)},\theta) - y^{(i)})^2$$

Parámetros

Considere además la siguiente elección de parámetros para la arquitectura:

$$\Theta = (\Theta^{(1)}, \Theta^{(2)}) \Longrightarrow \begin{aligned} \Theta_j^{(1)} &\sim \mathcal{N}(0, 1/n)^{d+1} \\ \Theta_j^{(2)} &\sim \mathcal{N}(0, 1/n) \end{aligned}$$

Con j que varía de 1 a n, con n el número de neuronas por capa.

Nos preguntamos entonces:

¿Cómo se ve, con una alta probabilidad, el gradiente $\nabla_\theta r(\Theta)$ dada esta inicialización?

¿Cómo se ve este gradiente?

Si restringimos el gradiente al componente $\theta^{(2)}$ podemos encontrar la siguiente relación entre la norma del gradiente y un kernel aleatorio con entradas alimentadas por el componente Θ y los vectores de características

$$\begin{aligned}\|\nabla_\theta r(\Theta)\|_2^2 &\geq \frac{4}{m^2}\left\|\sum_{i=1}^{m}\nabla_{\theta^{(2)}}\Phi(x^{(i)},\Theta)(\Phi(x^{(i)},\Theta)-y^{(i)})\right\|_2^2 \\ &= \frac{4}{m^2}\left((\Phi(x^{(i)},\Theta)-y^{(i)})_{i=1}^{m}\right)^T\bar{K}_\Theta(\Phi(x^{(j)},\Theta)-y^{(j)})_{j=1}^{m}\end{aligned}$$

Acto de fe…

En nuestro modelo simple podemos escribir la matriz como:

$$\bar{K}_\Theta = \sum_{k=1}^{n} v_k v_k^T \qquad \text{with} \quad v_k = \left(\varrho \left(\left\langle \Theta_k^{(1)}, \begin{bmatrix} x^{(i)} \\ 1 \end{bmatrix} \right\rangle \right) \right)_{i=1}^{m} \in \mathbb{R}^m, \quad k \in [n].$$

- Simétrica, semi-definida positiva.
- Entre más neuronas más probable que sea invertible. (crece rápido)
- El valor propio más pequeño crece linealmente respecto a n.

El control del gradiente

Con una alta probabilidad…

$$\|\nabla_\theta r(\Theta)\|_2^2 \geq \frac{4}{m^2}\lambda_{\min}(\bar{K}_\Theta)\|(\Phi(x^{(i)},\Theta)-y^{(i)})_{i=1}^m\|_2^2 \gtrsim \frac{n}{m}r(\Theta).$$

$$\mathbb{E}\left[\left(\frac{\partial^2\Phi(x,\Theta)}{\partial\theta_i^{(2)}\partial\theta_j^{(2)}}\right)^2\right]=0, \quad \mathbb{E}\left[\left(\frac{\partial^2\Phi(x,\Theta)}{\partial\theta_i^{(2)}\partial(\theta_j^{(1)})_k}\right)^2\right]\lesssim\delta_{i,j}, \quad \text{and} \quad \mathbb{E}\left[\left(\frac{\partial^2\Phi(x,\Theta)}{\partial(\theta_i^{(1)})_k\partial(\theta_j^{(1)})_\ell}\right)^2\right]\lesssim\frac{\delta_{i,j}}{n},$$

$$
\begin{aligned}
\|\nabla_\theta r(\Theta+\bar{\theta})\|_2^2 &\geq \frac{4}{m^2}\Big\|\sum_{i=1}^{m}\nabla_{\theta^{(2)}}\Phi(x^{(i)},\Theta+\bar{\theta})(\Phi(x^{(i)},\Theta+\bar{\theta})-y^{(i)})\Big\|_2^2 \\
&\underset{(5.5)}{=} \frac{4}{m^2}\Big\|\sum_{i=1}^{m}(\nabla_{\theta^{(2)}}\Phi(x^{(i)},\Theta)+\mathcal{O}(1))(\Phi(x^{(i)},\Theta+\bar{\theta})-y^{(i)})\Big\|_2^2 \\
&\underset{(*)}{\gtrsim} \frac{1}{m^2}(\lambda_{\min}(\bar{K}_\Theta)+\mathcal{O}(1))\|(\Phi(x^{(i)},\Theta+\bar{\theta})-y^{(i)})_{i=1}^m\|_2^2 \\
&\gtrsim \frac{n}{m}r(\Theta+\bar{\theta}),
\end{aligned}
$$

Conclusión de convergencia

- Dada nuestra inicialización podemos asegurar, y con alta probabilidad, que el gradiente en descenso converge a un ritmo exponencial a un punto de riesgo empírico arbitrariamente pequeño si tenemos un n suficientemente grande.
- Las iteraciones del algoritmo permanecen en un pequeño vecindario del punto de inicialización. Así, como los parámetros se mueven muy poco este tipo de entrenamiento es llamado entrenamiento perezoso.

En nuestro modelo simple mostramos la convergencia de SGD de forma controlada, esto debido a la sobre parametrización

Preguntas