Universidad Nacional de Colombia

Matemáticas del aprendizaje de máquinas 2022-I

¿Por qué la solución al problema de optimalidad funciona en las redes neuronales profundas?

Juan Lara, Ana Ramos, Ángel Giraldo y Jaime Zamora

Resumen:

En el presente documento exponemos posibles respuestas a la pregunta que da título al mismo, para ello realizamos una introducción teórica de los conceptos claves, presentamos los posibles problemas que se pueden presentar y finalmente exponemos tres posibles explicaciones dadas en [1] y [2]: observaciones sobre características de los puntos críticos de la función de pérdida, estudio de los caminos y conjuntos de nivel, y la simplificación de un caso para estudiar la convergencia del gradiente en descenso estocástico (SGD) [2].

. . . . . .

1. Introducción

En el mundo de la inteligencia artificial encontramos el aprendizaje de máquina y el aprendizaje profundo, estos campos nos permiten aprender ciertos sistemas dado un conjunto de datos [3]. En el proceso de aprendizaje la mayoría de las veces nos encontramos con que este depende directamente de un problema de optimización. Al momento de estudiar dichos problemas de optimización surgen diferentes inconvenientes que nos impiden explicar por qué en la práctica funcionan estos procesos de aprendizaje [1]; en el presente documento nos centramos en los inconvenientes que surgen al estudiar la naturaleza de los puntos a los cuales converge el método usado para resolver los problema de optimización descritos previamente (SGD) y permitir explicar sus buenos resultados.

1.1. Modelo general

En los procesos de apredizaje de inteligencia artificial encontramos el siguiente esquema general:

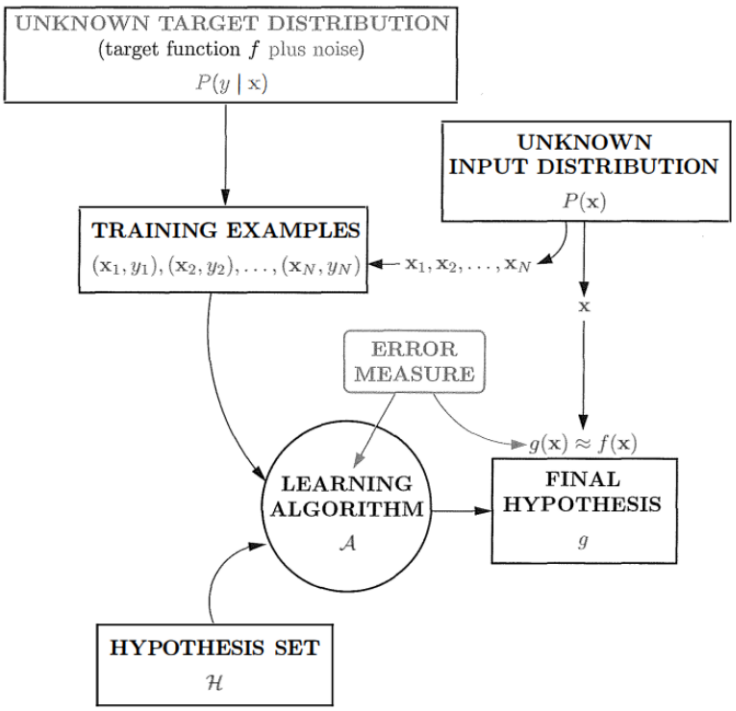

Figura 1: Esquema general

Entre el esquema es indispensable incluir una medida del error para los diferentes algoritmos de aprendizaje pues optimizar esta medida es lo que nos provee un *mejor* aprendizaje.

1.2. Función de pérdida y riesgo empírico

Formalmente para cuantificar qué tanto se aproxima una hipótesis g a la función objetivo f, usamos una medida del error que se conoce como la función de pérdida o *loss function* que denotamos $\mathcal{L}(f_s, z)$, esta función penaliza la mala clasificación o aproximación de un ejemplo particular z por una función f_z en el espacio de hipótesis, esto para un algoritmo entrenado con un conjunto de datos s,
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

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

Esta resulta ser una función importante pues nos permite dar una medida del error de una función del conjunto de hipótesis con lo cual podemos comparar estas y dado el algoritmo escoger una adecuada. El proceso de escoger una función adecuada dada una medida del error por lo general se realiza mediante un proceso de optimización de una función, la cual puede ser por ejemplo la función de riésgo empírico.

1.3. Gradiente en descenso

De entre todos los posibles métodos que pueden ser utilizados para minimizar una función de riesgo empírico encontramos los basados en el gradiente. Las razones principales por las cuales el gradiente en descenso es tan usado en este tipo de problemas son la precisión y eficiencia que tienen las redes FC para calcular derivadas puntuales por medio del algoritmo de backpropagation [2]. De esta forma se intenta minimizar el riesgo empírico mediante la actualización de los parámetros θ con la aplicación del algoritmo de gradiente en descenso estocástico.

Por su parte, el gradiente en descenso es un algoritmo de optimización iterativo de primer orden para encontrar un mínimo local de una función diferenciable, en este donde iniciamos en un punto al azar en el dominio de una función f y luego construimos un camino en pequeños pasos en dirección opuesta al gradiente de la función dada, en principio corresponde a lo siguiente:

```
While True:
    grad = evaluate_gradiente(J, corpus, theta)
    theta = theta - alpha * theta_grad
```

Sobre una función definida de $\mathbb{R}^2$ a $\mathbb{R}$ podemos visualizar el camino que genera el algoritmo en la figura 8.

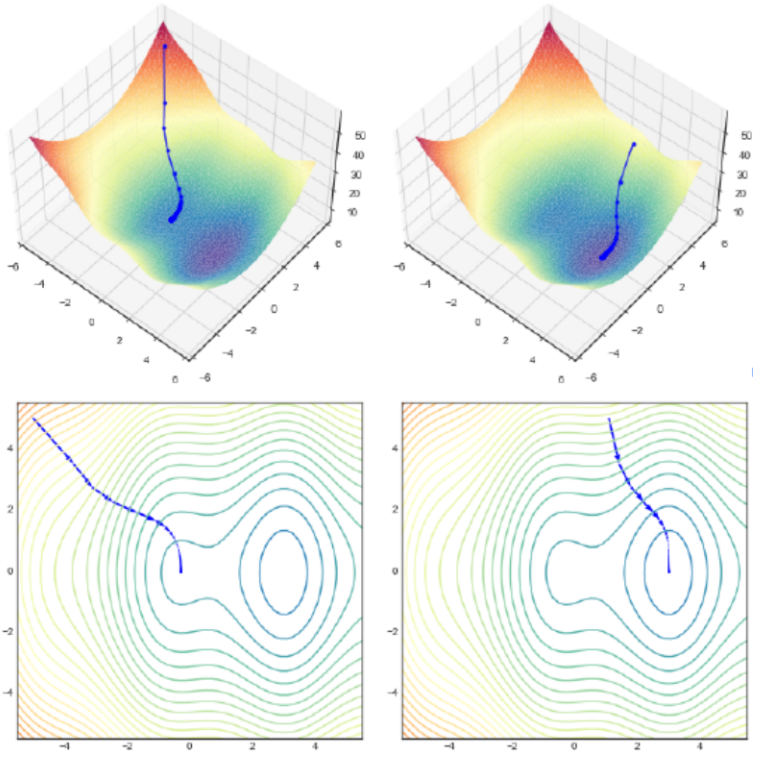

Figura 2: Gradiente en descenso

1.4. Gradiente en descenso estocástico

En la práctica no se usa gradiente en descenso, en cambio se usa el gradiente en descenso estocástico que es un método iterativo para optimizar una función objetivo con propiedades de suavidad adecuadas (por ejemplo, diferenciable o subdiferenciable). Puede considerarse como una aproximación estocástica de la optimización del descenso del gradiente, ya que reemplaza el gradiente real (calculado a partir de todo el conjunto de datos) por una estimación del mismo (calculado a partir de un subconjunto de datos seleccionado al azar). Especialmente en problemas de optimización de alta dimensión , esto reduce la carga computacional muy alta., logrando iteraciones más rápidas en el comercio para una tasa de convergencia más baja. En principio corresponde a lo siguiente:

```
While True:
    window = sample_window(corpus)
    grad = evaluate_gradiente(J, window, theta)
    theta = theta - alpha * theta_grad
```

Sobre una función definida de $\mathbb{R}^2$ a $\mathbb{R}$ podemos visualizar el camino que genera el algoritmo en la figura 3.

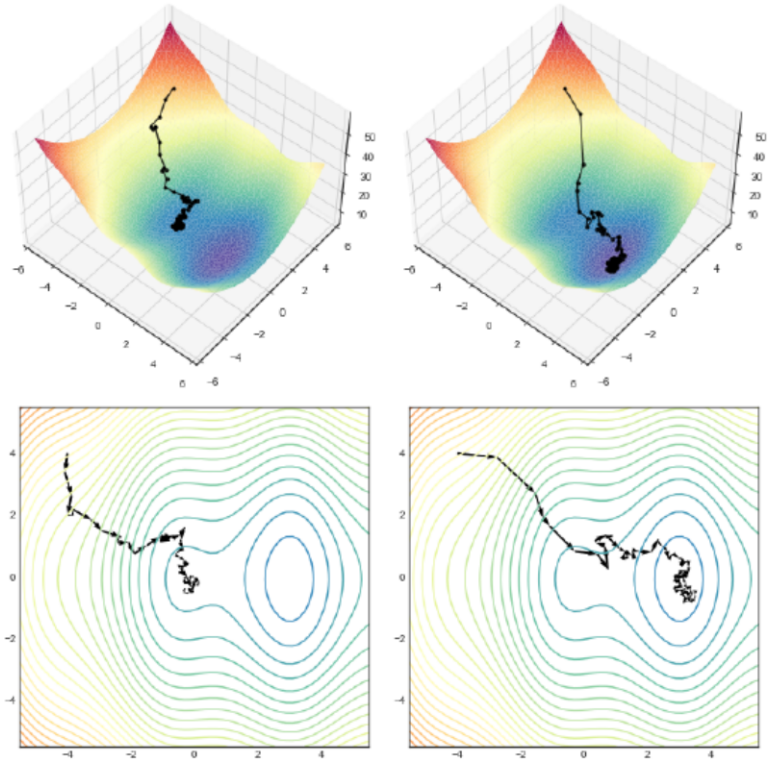

Figura 3: Gradiente en descenso estocástico

1.5. Convexidad

En el contexto de la teoría de la optimización es fundamental hablar sobre la convexidad de los problema existen diversos resultados que garantizan soluciones óptimas y eficientes en los problemas convexos [4].

Definición: Decimos que una función $f : \mathcal{D} \to \mathbb{R}$ es convexa si y solo si $\mathcal{D}$ es convexo (como conjunto) y para todo $\theta \in [0, 1]$ si $x, y \in \mathcal{D}$ se cumple

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

Luego, decimos que un problema de optimización es convexo si su función objetivo y la región factible son convexas. La principal razón por la que introducimos los problemas convexos es porque sabemos cómo resolverlos, más aún en [3] se enuncian diversos resultados que implican que en problemas convexos el gradiente en descenso estocástico se comporta bien, es decir, converge a la solución óptima de manera eficientemente. Por lo anterior es necesario estudiar la convexidad de los problemas de optimización asociados a las redes neuronales, para ello consideremos un ejemplo.

1.5.1. Ejemplo

Consideramos la red neuronal con función de activación es ReLu:

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

donde $\varrho_R(x) = \text{máx}\{0, x\}$. Observamos que para los parámetros

$$\theta = (1, -1, 1, 1, 1, 0), \qquad \bar{\theta} = (-1, 1, 1, 1, 0, 1)$$

obtenemos que la red neuronal funciona igual, es decir que al reemplazar los parámetros tenemos $\Phi(x, \theta) = \Phi(x, \bar{\theta})$, adicional, si consideramos el conjunto de datos $s = \{(-1, 0), (1, 1)\}$ entonces al calcular el riésgo empírico obtenemos

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

Pero al calcular este con el promedio aritmético de los parámetros, es decir con $\frac{\theta+\bar{\theta}}{2} = (0, 0, 1, 1, 1/2, 1/2)$ obtenemos $\hat{\mathcal{R}}_s(\Phi(\cdot, \frac{\theta+\bar{\theta}}{2})) = \frac{1}{2}$ puesto que $\Phi(\cdot, \frac{\theta+\bar{\theta}}{2}) = 0$ lo cual muestra que nuestra función $\Phi(\cdot, \theta)$ no es convexa.

En general la función de riesgo empírico asociada a una red neuronal no es una función convexa [2], esto representa un problema pues no hay garantía teórica de obtener la solución óptima a un problema de optimización no convexo [1] y, dado el buen funcionamiento de las redes neuronales en la práctica [5] es necesario entender por qué la solución numérica a la cual llegamos es buena, es decir,

2. Problemas

Debido a la no-convexidad del problema mencionada anteriormente, surgen inconvenientes con el SGD: pueden existir múltiples mínimos subóptimos, la función objetivo puede tener puntos de silla en los cuales el Hessiano se anula, o incluso si no fuese el caso de que haya mínimos subóptimos, podría haber áreas del espacio de parámetros en donde el gradiente es muy pequeño, con lo cual salir de dichas regiones puede ser demorado para el algoritmo [2]. Con respecto al problema de los múltiples mínimos subóptimos, en [5], [6] se muestra que dicha situación se presenta en situaciones comunes de aprendizaje. También se muestra para redes de tamaño fijo en [7], [8] que en general el conjunto de las redes neuronales es no convexo y no cerrado, lo cual nos dice que el método SGD tiene problemas con llegar a mínimos óptimos. Adicional a esto, en varias situaciones se ha mostrado [9], [10] que el problema de hallar el mínimo global del problema de optimización es un problema NP-duro. En la figura a continuación vemos la proyección bidimensional del paisaje de una función de pérdida, esto es, la proyección del gráfico de la función

$$\theta \to \hat{\mathcal{R}}_s(\Phi(\cdot, \theta)).$$

Se hace evidente de la representación que el problema tiene más de un mínimo:

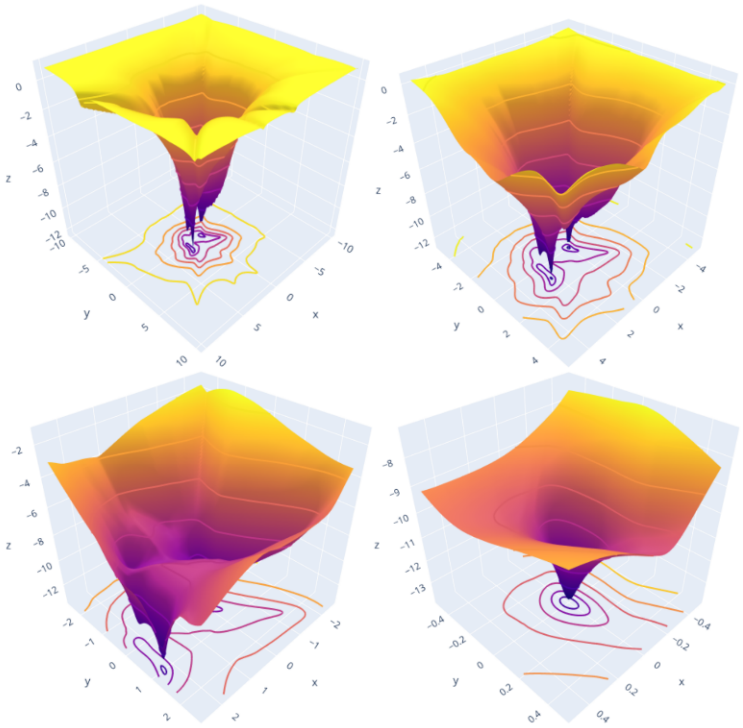

Figura 4: Proyección del paisaje de pérdida de una red con 4 capas y función de activación ReLu sacada de [2]

En vista de lo anterior, el marco teórico clásico no funciona para explicar el funcionamiento

el la práctica del aprendizaje profundo. De hecho, en [11], el estado del arte del aprendizaje profundo en 1998 se resumía en que: "No hay fórmula para garantizar que la red converge a una buena solución, la convergencia es rápida, o siquiera si ocurre convergencia en lo absoluto."
Sin embargo, en la práctica se observa convergencia bastante a menudo, aunque no hay sustento o justificación teórica de porqué ocurre. Es por eso que en la Sección 3 recopilamos 3 posibles explicaciones de porqué y cuándo hay convergencia garantizada teóricamente.

3. Posibles explicaciones

Basados en los principales resultados de [1] y [2] presentamos explicaciones que nos permiten entender el comportamiento de los mínimos locales de nuestras funciones objetivo en los problemas de optimización asociados a las redes neuronales, esto nos permite comprender porqué a pesar de los problemas ya mencionados obtenemos buenos resultados en la práctica [5] y así mismo dar respuesta al título del documento.

3.1. Puntos críticos de la función de pérdida

La inestabilidad de un punto crítico α de una función diferenciable f significa que para los puntos cercanos que se encuentran en cierta dirección de α, menos el gradiente de la f evaluado en α no "apunta"hacia α partiendo desde estos puntos. Es decir, menos el gradiente diverge de α en esta dirección. Cada valor propio negativo de la matriz Hessiana de la función evaluada en α significa una dirección en la que esto ocurre. Por lo tanto, si tenemos un algoritmo que da pasos en la dirección de menos el gradiente de la función de pérdida evaluado en el punto actual, como es el caso de GD o SGD, al acercarse este a un punto crítico es más probable que no converja hacia él entre más inestable sea.

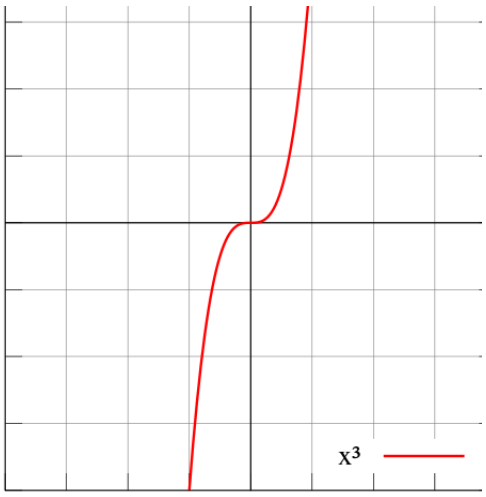

Figura 5: La función $f = x^3$ presenta un punto crítico inestable en $x = 0$.

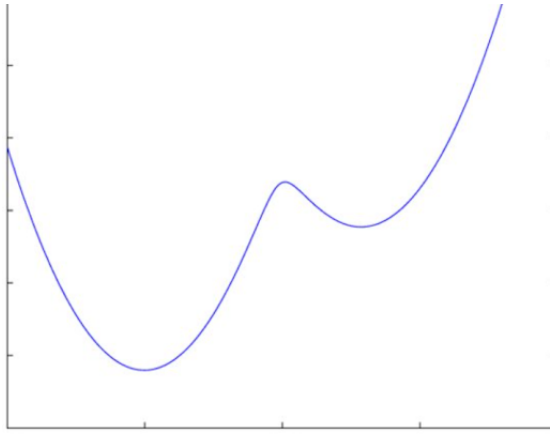

Figura 6: Función con mínimos locales estables, y un máximo local inestable.

Algunos resultados provenientes de la física estadística y de la teoría de matrices aleatorias, y en particular, algunas propiedades del Hamiltoniano de un modelo de vidrio de espín, han permitido notar y explicar ciertas características de la estabilidad de los puntos críticos de la función de pérdida de redes neuronales profundas [12].

3.1.1. Herramientas de la física estadística

De manera simplista, podemos entender el Hamiltoniano como un operador o función que envía a un sistema físico, descrito por ciertas propiedades o variables, a su energía total. A un modelo de vidrio de espín, por otro lado, lo podemos entender como un sistema magnético en el que los momentos magnéticos (intensidad y dirección de ciertos campos magnéticos), son aleatorios.

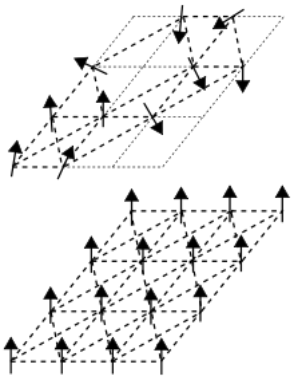

Figura 7: Modelo de vidrio de espín

Resulta que el Hamiltoniano de un vidrio de espín tiene la propiedad de que con alta probabilidad, el porcentaje de valores propios negativos de la matriz Hessiana de la función de pérdida evaluada en un punto crítico, es más alto entre más lejano esté este punto crítico del mínimo global. Es decir, se tiene que con alta probabilidad, entre más lejano esté un punto

crítico del mínimo global más inestable es. Del mismo modo, con alta probabilidad un punto crítico cercano al mínimo global será estable.

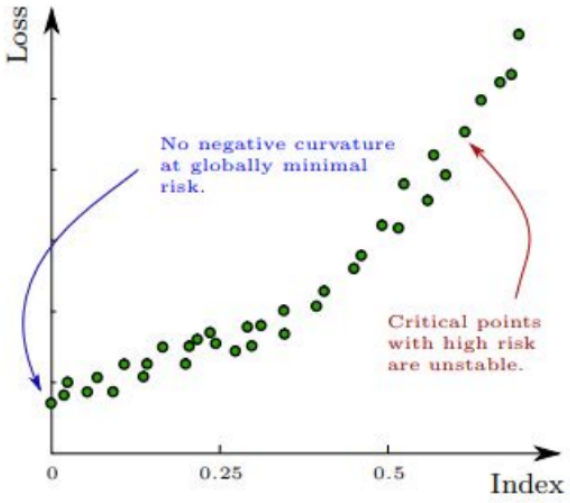

Figura 8: Gradiente en descenso

En [12] se prueba que, bajo ciertas condiciones, la función de pérdida de una red neuronal profunda puede ser considerado como el Hamiltoniano de un modelo de vidrio de espín. Estas condiciones difícilmente son cumplidas por las funciones de pérdida de las redes neuronales en la práctica, pero aún así, las anteriores características sobre la estabilidad de los puntos críticos sí se suelen tener [2].

3.1.2. Resultados generales

En [12] también se muestra que bajo algunas condiciones sobre los datos de entrenamiento y los parámetros de esta función, entre más capas se le agreguen a una red neuronal, los mínimos locales tienden a tener valores más cercanos al mínimo global. Algunos resultados relacionados se han demostrado para un caso más general, sin suponer condiciones sobre la distribución o inicialización de los parámetros, entre otras que se habrían hecho anteriormente. Por ejemplo, en [13] se muestra que para redes que cumplen condiciones más generales y que son lo suficientemente grandes, se garantiza que los mínimos locales que son estrictamente menores a los otros puntos en una vecindad lo suficientemente pequeña, no existen.

3.2. Caminos y conjuntos de nivel

La interpretación previa relacionada a modelos de vidrio de espín realiza importantes simplificaciones en cuanto a la naturaleza no lineal del modelo [12]. Ante ello, otra posible aproximación (que no realiza simplificaciones en cuanto a la no linealidad) a la explicación

de la convergencia, se encuentra en el estudio de los caminos en el espacio de parámetros para los cuales el riesgo empírico asociado es no-creciente a lo largo del camino. Claramente, de existir caminos a lo largo de los cuales el riesgo empírico es no-creciente, que vayan desde un punto arbitrario al mínimo global, podríamos garantizar la no existencia de mínimos no globales.
A continuación veremos un resumen del desarrollo realizado en [14], el cual, entre otras, centra su principal resultado sobre la topología en las redes neuronales de una única capa, sin asumir simplificaciones sobre la no-linealidad del modelo.
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.
Con esto podemos pasar a definir los conjuntos de nivel de $F(\theta)$ como

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

Estudiar los conjuntos de nivel nos permite tener mejor entendimiento de la estructura de los puntos críticos de de $F(\theta)$. En particular, nos permite saber si hay mínimos locales que no sean globales, dado que sepamos si $\Omega_F(\lambda)$ es conexo en cada nivel de energía λ:

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 si $\kappa = 0$ y también si $\kappa > 0$ para $K = 2$ [15]. Más aún, el resultado afirma que cada θ se puede conectar al nivel de energía más bajo mediante un camino estrictamente decreciente. Este resultado es profundo en el sentido de que introducir regularización cambia drásticamente la topología del modelo.

Lo anterior nos muestra que los modelos lineales no tienen problema, pero queremos ver qué

pasa en el caso en que tengamos un modelo de la forma

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

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 [14].
A pesar de lo anterior, aún se puede recuperar (hasta cierto punto) la conexidad si se permite un incremento pequeño en el nivel de energía. El resultado más importante de [14] dice que la cantidad que aumenta la energía está acotada por una cantidad que representa un *trade-off* entre la sobreparametrización del modelo y la suavidad de la distribución de los datos. Este resultado se prueba para una red con una única capa intermedia y la regularización $\mathcal{R}(\theta) = \|\theta\|_1$.
El resultado considera un escenario no asintótico dado por una cantidad fija m de neuronas 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 $\text{máx}(\lambda, \epsilon)$, donde ϵ decrece de acuerdo a la sobreparametrización del modelo.
De dicho resultado 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.

3.3. Convergencia del SGD

En el marco de las redes neuronales sobreparametrizadas podemos encontrar que los parámetros aleatorios con los que se inicializa la red cambian poco en el proceso de entrenamiento, y este fenómeno puede explicarse junto a la convergencia probable del gradiente en descenso para redes profundas suficientemente sobreparametrizadas.

3.3.1. Ejemplo

Una red de arquitectura $((1, n, n, 1), \rho)$ (2 capas ocultas con n neuronas) puede ser entrenada para encontrar una función que interpole 4 puntos en el plano cartesiano. ¿Que puede observarse si tomamos valores de n suficientemente grandes?

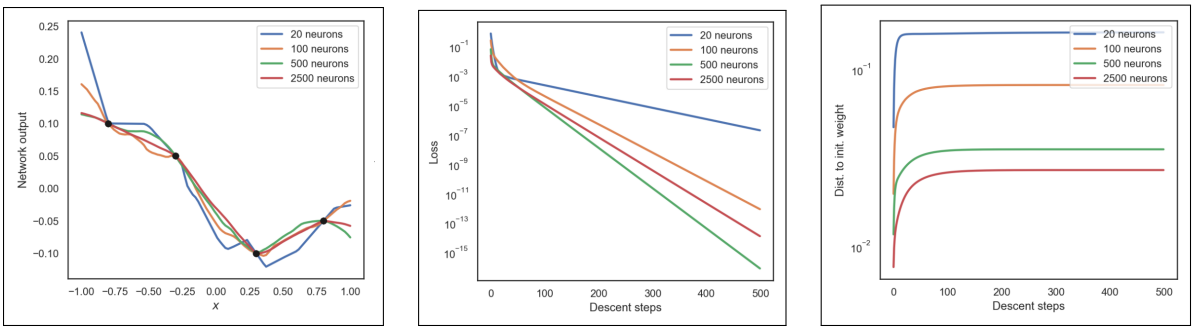

Figura 9: Fenómeno de sobreparametrización

A grandes rasgos puede decirse que dada una inicialización de los parámetros estos permanecen cerca a medida que avanza el proceso de entrenamiento.

3.3.2. Cota del gradiente

Para entender este comportamiento consideremos una NN de dos capas sin bias de la forma:

$$\Phi(x,\theta) := \sum_{j=1}^{n} \theta_j^{(2)} \rho(\langle \theta_j^{(1)}, (x,1)^T \rangle)$$

Donde cada $\theta_j^{(1)} \in \mathbb{R}^{d+1}$ y $\theta^{(2)}$ es un vector de $\mathbb{R}^n$. Para la pérdida cuadrática y datos de entrenamiento $s = ((x^{(i)}, y^{(i)}))_{i=1}^m$ el riesgo empírico esta dado por:

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

Ahora necesitamos establecer de que forma se inicializan nuestros parámetros. Si $\Theta = (\Theta^{(1)}, \Theta^{(2)})$ entonces

$$\Theta_i^{(1)} \sim N\left(0, \frac{1}{n}\right)^{d+1}, \quad \Theta_i^{(2)} \sim N\left(0, \frac{1}{n}\right)$$

Dada esta elección para Θ podemos entender como se ve el gradiente $\nabla_\theta r(\Theta)$ con una alta probabilidad. Y es que al restringir el gradiente a $\theta^{(2)}$ vemos que

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

Es decir que la magnitud del gradiente esta acotado inferiormente por una relación entre el vector con el error dentro el conjunto de datos y un kernel con entradas aleatorias, similar al kernel tangencial que aplicamos a los vectores de características x_i y que depende también de nuestros parámetros Θ. El kernel K_Θ contiene la información relevante sobre la convergencia

de SGD. Y es que con una alta probabilidad, con un número de neuronas n suficientemente grande, tenemos que esta matriz es definida positiva y además su valor propio más pequeño crece linealmente con n. Concluimos entonces que con una alta probabilidad la magnitud del gradiente tiene un orden superior a un múltiplo por escalar de la función de riesgo empírico, más precisamente:

$$||\nabla_\theta r(\Theta)||_2^2 \succeq \frac{n}{m} r(\Theta)$$

Es posible extender esta cota para el caso en que nos encontramos en una vecindad de Θ y obtenemos que

$$||\nabla_\theta r(\Theta + \theta)||_2^2 \succeq \frac{n}{m} r(\Theta + \theta)$$

Esto quiere decir que con una alta probabilidad, dada la inicialización de Θ, en una bola de radio fijo al rededor de Θ la magnitud del gradiente del riesgo empírico esta acotado inferiormente por $\frac{n}{m}$ veces el riesgo empírico. Más aún, este resultado nos permite afirmar que para pasos suficientemente pequeños y una cantidad suficiente de neuronas, el algoritmo SGD converge con una velocidad exponencial a un riesgo empírico arbitrariamente pequeño. Además los parámetros se mantienen en una vecindad no muy lejana del punto de partida, es decir que no cambian considerablemente durante la etapa de entrenamiento.

4. Conclusiones

En cada una de las explicaciones que abordamos en este documento se presenta coherencia con los resultados vistos en la práctica al usar redes neuronales [2]; en primer lugar vimos que surgen diversos problemas que dan sentido a la pregunta de este documento y resaltan la importancia de conocer su solución. Al estudiar la topología de los conjuntos de nivel de la función de pérdida podemos caracterizar la naturaleza de los puntos críticos de esta. Vimos que para las redes de tipo "shallowçon función de activación ReLu, tenemos cierto nivel de conexidad por medio de caminos a lo largo de los cuales el riesgo no incrementa mucho.
Finalmente, también vimos que, bajo ciertas asunciones, con un alta probabilidad, cerca al punto de mínimo global encontraremos los mínimos locales, y que lejos de este encontramos puntos críticos inestables. Además, bajo condiciones más generales, ciertos tipos de mínimos locales para redes lo suficientemente grandes no existen. Por consiguiente, el gradiente en descenso estocástico converge a buenas soluciones, pues incluso cuando no llega al valor óptimo se encuentra lo suficientemente cerca y por el comportamiento descrito de las funciones de riesgo empírico tenemos que la solución a la que se llega es buena, es decir, comete un error bajo.
Por lo anterior podemos explicar el buen comportamiento de las soluciones a las que llegamos en la práctica, dado que llegan a *buenos* parámetros en la solución al problema de optimización. Para profundizar más en el desarrollo de estas preguntas podemos ver [2], [6], [9]

Referencias

[1] R. Vidal, J. Bruna, R. Giryes y S. Soatto, "Mathematics of deep learning," en, dic. de 2017. DOI: 10.48550/arXiv.1712.04741. dirección: https://arxiv.org/abs/1712.04741v1 (visitado 21-06-2022).

[2] J. Berner, P. Grohs, G. Kutyniok y P. Petersen, "The modern mathematics of deep learning," *arXiv:2105.04026 [cs, stat]*, mayo de 2021, arXiv: 2105.04026. dirección: http://arxiv.org/abs/2105.04026 (visitado 20-06-2022).

[3] *Machine Learning*, es-co. dirección: https://www.ibm.com/co-es/analytics/machine-learning (visitado 25-06-2022).

[4] *Lectures on convex optimization*. New York, NY: Springer Berlin Heidelberg, 2018, ISBN: 9783319915777.

[5] I. Safran y O. Shamir, *Spurious Local Minima are Common in Two-Layer ReLU Neural Networks*, 2017. DOI: 10.48550/ARXIV.1712.08968. dirección: https://arxiv.org/abs/1712.08968.

[6] P. Auer, M. Herbster y M. K. K. Warmuth, "Exponentially many local minima for single neurons," en *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer y M. Hasselmo, eds., vol. 8, MIT Press, 1995. dirección: https://proceedings.neurips.cc/paper/1995/file/3806734b256c27e41ec2c6bffa26d9e7-Paper.pdf.

[7] P. Petersen, M. Raslan y F. Voigtlaender, *Topological properties of the set of functions generated by neural networks of fixed size*, 2018. DOI: 10.48550/ARXIV.1806.08459. dirección: https://arxiv.org/abs/1806.08459.

[8] J. Berner, D. Elbrächter y P. Grohs, "How degenerate is the parametrization of neural networks with the ReLU activation function?" *CoRR*, vol. abs/1905.09803, 2019. arXiv: 1905.09803. dirección: http://arxiv.org/abs/1905.09803.

[9] A. L. Blum y R. L. Rivest, "Training a 3-node neural network is NP-complete," *Neural Networks*, vol. 5, n.º 1, págs. 117-127, 1992, ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80010-3. dirección: https://www.sciencedirect.com/science/article/pii/S0893608005800103.

[10] J. S. Judd, *Neural Network Design and the Complexity of Learning*. Cambridge, MA, USA: MIT Press, 1990, ISBN: 0262100452.

[11] G. B. Orr y K.-R. Müller, eds., *Neural Networks: Tricks of the Trade*, ép. Lecture Notes in Computer Science. Springer, 1998, vol. 1524, ISBN: 3-540-65311-2. dirección: http://dblp.uni-trier.de/db/conf/nips/nips1996.html.

[12] A. Choromanska, Y. LeCun y G. Ben Arous, "Open Problem: The landscape of the loss surfaces of multilayer networks," en *Proceedings of The 28th Conference on Learning Theory*, P. Grünwald, E. Hazan y S. Kale, eds., ép. Proceedings of Machine Learning Research, vol. 40, Paris, France: PMLR, 2015, págs. 1756-1760. dirección: https://proceedings.mlr.press/v40/Choromanska15.html.

[13] B. D. Haeffele y R. Vidal, "Global Optimality in Neural Network Training," en *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, págs. 4390-4398. DOI: 10.1109/CVPR.2017.467.

[14] C. D. Freeman y J. Bruna, "Topology and Geometry of Half-Rectified Network Optimization," 2016. DOI: 10.48550/ARXIV.1611.01540. dirección: https://arxiv.org/abs/1611.01540 (visitado 27-06-2022).

[15] A. M. Saxe, J. L. McClelland y S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," 2013. DOI: 10.48550/ARXIV.1312.6120. dirección: https://arxiv.org/abs/1312.6120 (visitado 27-06-2022).