# GEOEMTRIA STRUCTURILOR REŢELELOR NEURALE

Vlad Raul Constantinescu[1], Ionel Popescu[2,3]

[1] Şcoala doctorală a Facultăţii de Matematică, Universitatea din Bucureşti
vlad_constantinescu95@yahoo.com
[2] Facultatea de Matematică şi Informatică, Universitatea din Bucureşti
[3] Institul de Matematică al Academiei Române "Simion Stoilow"
ioionel@gmail.com

O formulare matematică a problemei de învatare automată este următoarea. Presupunem că avem un set de date $(X_i, Y_i)_{i=\overline{1,N}}$ extrase independent după o distribuţie $\mu$, unde $X_i$ sunt datele de input iar $Y_i$ sunt etichetele asociate. De exemplu, daca avem un set de imagini, avem $X_i$ este imaginea în sine iar $Y_i$ este descrierea imaginii (câine, pisică, avion, etc). În acest context problema este aceea de a determina o funcţie $f : \mathbb{R}^n \to \mathbb{R}^m$ în asa fel încât să avem $Y_i = f(X_i)$. Este un fel de problema de regresie. În mod realist este dificil de găsit o astfel de funcţie datorituă multor inconveniente, de exemplu, avem zgomot, sau o asftel de funcţie este mult prea complicată. Asa că în general se caută o aftel de funcţie într-o familie destul de simplă. Pe de altă parte, cum ziceam, nu ne putem aştepta ca $f$ să descrie foarte bine datele, asa că trebuie să definim un fel de eroare asociată. In acest context se defineşte o funcţie de pierdere, $L(Y', Y)$ care masoară cat de departe este $Y'$ de $Y$, unde $Y = f(X)$ este predicţia şi $Y'$ este eticheta reală a lui $X$. O astfel de funcţie la îndemână este $L(Y', Y) = \|Y - Y'\|^2$, însă sunt şi alte funcţii care joaca un rol mai bun în astfel de situaţii, cum ar fi entropia relativă pentru cazul în care $Y, Y'$ iau doar valori discrete şi sunt probabilităţi.

Reţelele neurale standard determină o parametrizare a acestor funcţii dată de structura retelei. Putem intepreta aceasta ca o funcţie de la un spatiu $\mathbb{R}^k$ în spaţiul de funcţii $\mathcal{F}(\mathbb{R}^n; \mathbb{R}^m)$. Notăm această funcţie cu $V : \mathbb{R}^k \to \mathcal{F}(\mathbb{R}^n; \mathbb{R}^m)$. Pentru o pereche $(X, Y) \in \mathbb{R}^n \times \mathbb{R}^m$ si folosind funcţia $L$ de mai sus, putem defini acum $F_{(X,Y)}(\theta) = L(V(\theta)(X), Y)$ care ia valori reale. Având la îndemână un set de date $(X_i, Y_i)_{i=\overline{1,N}}$, în principiu vrem să optimizăm (minimizăm mai precis) după funcţia pierdere medie de forma

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} F_{(X_i,Y_i)}(\theta).$$

În principiu noi am dori de fapt minimizarea riscului mediu data de $\mathbb{E}[F_{(X,Y)}(\theta)]$, unde media este calculată în raport cu măsura de probabilitate $\mu$ ce generează datele. Din păcate această distribuţie nu este cunoscută, însă putem aproxima această medie cu varianta discretă de mai sus.

Metoda standard de a găsi $\theta$ optim pentru funcţia $\mathcal{L}$ este bazată pe metoda *gradient descent*. Această metodă constă în găsirea unei linii poligonale cu direcţia opusă gradientului lui $\mathcal{L}$ care să ne conducă spre un mimim local. Asta implică aşadar calculul gradientului lui $\mathcal{L}$ şi în particular pentru fiecare $F_{(X_i,Y_i)}$. Acesta din păcate este foarte costisitor, aşa că însăşi problema calculului gradientului lui $\mathcal{L}$ devine intractabil. Modificarea standard aplicată este cea de mini batch, adică în loc să calculăm la fiecare pas $\mathcal{L}$ si gradientul asociat, calculăm gradientul doar de-a lungul unui subsample mai mic din datele disponibile. Aşa că avem acum de calculat doar un numar mai mic de date. Aceasta metoda este cunoscută ca metoda *stochastic gradient descent* pentru că alegem de fiencare dată un subsample mai mic şi la fiecare pas ne miscăm în direcţia acestui gradient generat de subsample.

Pentru reţelele neurale, înca nu există o justificare suficientă pentru marele mister, anume faptul că aceste metode sunt foarte eficiente din punct de vedere practic.

Una din problemele pe care le propunem pentru această teză este studierea proprietăţilor acestor funcţii de risc $F_{(X;Y)}(\theta)$. In particular am dori să vedem ce proprietăţi intrinsece trebuie să aibă aceste clase de funcţii ce garantează metoda gradientului să funcţioneze aşa de bine.

Una din direcţiile de studiu este legată de flexibilizarea structurilor de reţele neurale şi studiul mai întâi empiric ale acestor reţele de tip neural. De exemplu, putem în loc de reţele de tip standard să gândim nişte reţele cu o topologie de tip diferit. Mai exact spus, reţelele clasice sunt de tipul stratul 0 citeşte datele şi următorul strat este compus din termeni de stratul strict de dinainte. În acest fel topologia este oarecum liniară. Există şi alte structuri ceva mai complexe, în care de exemplu straturile interactionează între ele într-un anumit fel. Ce ne propunem aici este analiza structurilor şi topologiei asociate. Un exemplu simplu în acest caz ar fi cel în care straturile sunt dispuse pe un tor şi reteaua neurală este organizată pe un mesh pe tor. In acest fel stratul de input si de output constând într-un numar de noduri alese aleator pe acest mesh.

O altă direcţie de lucru este cea în care vrem să investigăm ce proprietăţi, sa le numim $P$, sunt importante pentru aceste reţele neurale. De exemplu, o primă abordare ar fi cea în care să studiem ce proprietăţi se păstrează în restricţiile acestor funcţii de tip $F_{(X,Y)}$ dea lungul unor drepte şi plane luate aleator în $\mathbb{R}^k$ (spaţiul ambiental al parametrilor). Este foarte posibil ca acest tip de strutură aleatoare văzută doar la nivel de restricţii pe structuri geometrice simple să aibă ceva în comun. Urmatorul pas în această analiză ar fi cel legat de proprietăţile ce se păstrează la adunare, adică date fiind două astfel de funcţii cu proprietatea $P$, suma lor să păstreze aceste proprietăţi. O astfel de proprietate este foarte importantă deoarece ne poate furniza diverse alte structuri potenţial generale care să reproducă reţelele neurale însa într-o formă mult mai simplă şi în acelaşi timp cu o justificare matematică riguroasă.

## Bibliografie


[1] Omer Bobrowski, Matthew Kahle, Primoz Skraba, et al. Maximally persistent cycles in random geometric complexes. *The Annals of Applied Probability*, 27(4):2032–2060, 2017

[2] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016

[3] Qianli Liao and Tomaso Poggio. Theory ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*, 2017.

[4] Tomaso Poggio, Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh Mhaskar. Theory of deep learning iii: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*, 2017.

[5] Christian Robert. Machine learning, a probabilistic perspective, 2014.

[6] Jurgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[7] Herut Uzan, Shira Sardi, Amir Goldental, Roni Vardi, and Ido Kanter. Biological learning curves outperform existing ones in artificial intelligence algorithms. *Scientific reports*, 9(1):1–11, 2019.

[8] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2019.

[9] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Brando Miranda, Noah Golowich, and Tomaso Poggio. Theory of deep learning iib: Optimization properties of sgd. *arXiv preprint arXiv:1801.02254*, 2018.