

Laborator 8

Lucian M. Sasu

1 Perceptron multistrat

Implementați algoritmul Backpropagation pentru o rețea feedforward multistrat (MLP), pentru o problemă de clasificare. Se rezolvă problema de recunoaștere optică a caracterelor, folosind setul de date MNIST descris la <http://yann.lecun.com/exdb/mnist/>; setul de date este același ca la laboratorul 4.

MLP este prezentat în [1], pag. 57 și următoarele și în [2].

Funcția de activare folosită pentru neuronii din stratul ascuns poate fi sigmoida logistică sau tangenta hiperbolică. Neuronii din stratul de ieșire folosesc funcția de activare softmax, funcția de eroare este cross entropy.

Se va folosi regularizarea pentru ponderile legăturilor dintre noduri; ponderile asociate *bias*-urilor nu se regularizează. Se poate face învățare în stil gradient descent, minibatch sau batch (offline); recomandăm minibatch. Se vor face mai multe epoci de antrenare, până când modificarea ponderilor și a valorilor de bias scade sub un prag.

Inițializare valorilor ponderilor și a coeficienților de bias se face cu valori aleatoare mici în jurul lui 0; puteți opta de asemenea pentru strategia Glorot [3] sau He [4].

Scorul se calculează ca acuratețe pe setul de testare din MNIST. Opțional, se poate reprezenta matricea de confuzie pentru setul de testare.

Se cere implementarea următoarelor două puncte:

1. Rețea neurală cu un singur strat ascuns, a se vedea figura 6.3 din curs; numărul de noduri de intrare este 784, numărul de neuroni din stratul ascuns se fixează apriori, e.g. 100, numărul de neuroni de ieșire este numărul de clase; antrenare se face pe tot setul de antrenare (60000 linii), testarea pe setul de testare. Funcția de activare pentru neuronii din stratul de ieșire este softmax, în rest tangenta hiperbolică. Se raportează acuratețea de clasificare.

2. Numărul de neuroni din stratul ascuns se determină prin **metoda hold-out**: setul de antrenare se partiționează în primele 50000 linii pentru antrenare, restul pentru validare; pentru număr de neuroni din stratul se consideră o mulțime de valori candidat, de exemplu: $\{10, 50, 100, 200, \dots\}$. Se face antrenare folosind doar primele 50000 de date, folosind mai multe epoci (condiția de oprire este dată mai sus), după care se face testarea pe setul de validare. Numărul de neuroni din stratul ascuns pentru care eroarea pe setul de validare este cea mai mică se consideră a fi optimal și se folosește pentru a face antrenarea pe toate cele 60000 linii din setul de antrenare inițial (incluzând deci liniile din setul de validare); în final se raportează acuratețea pe setul de testare.

Opțional:

1. Se poate face **k-fold cross validation**, pentru a obține o evaluare mai veridică a performanțelor modelelor candidat. Un model candidat are o valoare anume pentru numărul de neuroni din stratul ascuns. Se va lua $k = 10$.
2. Se poate face validare pentru rata de învățare.

Termen de predare: săptămâna 7-11 mai.

Bibliografie

- [1] Lucian M. Sasu, *Sisteme computationate inteligente — note de curs*, [link github](#)
- [2] Michael Nielsen, *Neural Networks and Deep Learning*, <http://neuralnetworksanddeeplearning.com/>
- [3] Xavier Glorot, Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks*, Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics, 2010
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015