

Uniwersytet Bielsko-Bialski

Wydział Budowy Maszyn i Informatyki



MK_LAB

Zadanie polegało na nauczaniu sieci neuronowej wykonywania odejmowania dwóch liczb binarnych. Sieć dostaje na wejściu dwa bity (po jednym z każdej liczby) i musi wypuścić bit wyniku. Haczyk polega na tym, że musi "pamiętać" pożyczkę z poprzedniego kroku - dokładnie tak jak my robimy odejmowanie pisemne.

Parametry wariantu 11:**

- Liczby 15-bitowe (maksymalna wartość: 16383)
- 2000 próbek treningowych
- Operacja: odejmowanie ($A - B$, gdzie $A \geq B$)

Generujemy losowe pary liczb i zamieniamy je na bity. Ważne: bity są odwrócone (najpierw najmniej znaczący), bo tak łatwiej się liczy od lewej do prawej.

Przykład:

```
...
x1: 1001110000000000 14369
x2: - 0110100000000000 2908
-----
t: = 0011010000000000 11461
...
```

Uczenie sieci

Używamy algorytmu BPTT (Backpropagation Through Time) - czyli propagujemy błąd wstecz przez wszystkie 15 kroków czasowych.

Optymalizator: **RMSProp + Nesterov momentum**

- Learning rate: 0.05
- Momentum: 0.80
- Minibatch: 100 próbek
- Epoki: 5

Wyniki

Strata spada z ~ 0.69 (losowe zgadywanie) do praktycznie 0 po około 50 iteracjach.

Przykład 1:

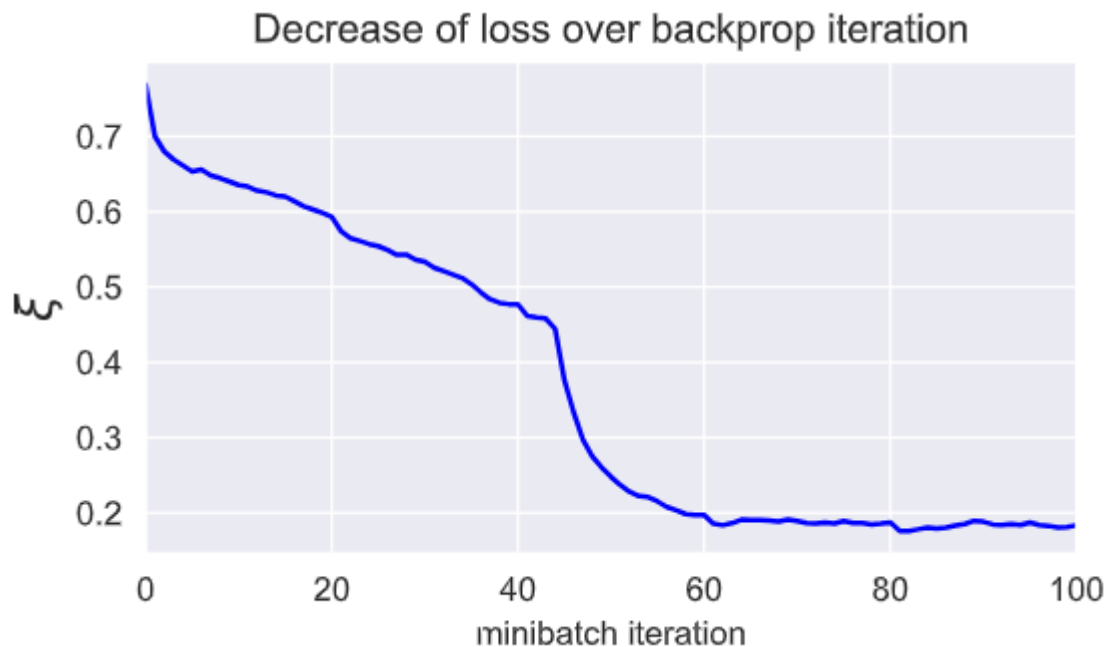
```
x1: 1011000110000000 6291
x2: - 0000100100000000 1156
t: = 1010100010000000 5135 ← cel
y: = 1010100010000000 ← predykcja ✓
```

Przykład 2:

```
x1: 0110111001000000 9582
x2: - 1100001000000000 2147
```

t: = 101010000100000 7435 ← cel
y: = 101010000100000 ← predykcja ✓
...

****Dokładność: 100%**** - sieć nauczyła się prawidłowo obsługiwać mechanizm pożyczki.



Powyższy rysunek pokazuje, że trening doprowadził do utraty 0. Oczekujemy, że sieć nauczyła się doskonale wykonywać dodawanie binarne dla naszych przykładów szkoleniowych. Jeśli przepuścimy przez sieć kilka niezależnych przypadków testowych i wydrukujemy je, zobaczymy, że sieć również wygeneruje prawidłowe dane wyjściowe dla tych przypadków testowych.

