sages

Czym jest głębokie uczenie?

(ang. deep learning)

sages

Historia

Trzy fale popularności sieci neuronowych

Pierwsza fala (1943-1969)

- 1943 — pierwszy model neuronu (McCulloch i Pitts)
  - dwie kategorie obiektów,
  - sprawdzał, czy funkcja *f(x,w)* jest dodatnia, czy ujemna,
  - wagi były ustawiane ręcznie,
- 1958-1962 — pierwsze modele uczące się wag na podstawie danych (Rosenblatt, Widrow i Hoff),
- krytyka tego podejścia ze względu na **brak możliwości nauczenia się funkcji XOR**
  - f(0, 0) = f(1, 1) = 0
  - f(1, 0) = f(0, 1) = 1
- sieci neuronowe zostały porzucone na ponad dekadę.

Druga fala (1986-1999)

- 1986 — udane zastosowanie **wstecznej propagacji** w uczeniu się sieci neuronowych (Rumelhart, Hinton, Williams),
- 1986 — pojęcie **rozproszonej reprezentacji** (Hinton),
  - dane wejściowe powinny być reprezentowane przez wiele cech,
  - każda cecha powinna reprezentować możliwie wiele danych wejściowych,
- 1997 — sieć z długą pamięcią krótkoterminową (**LSTM**) do modelowania danych sekwencyjnych (Hochreiter i Schmidhuber),
- koniec lat 90-tych XX wieku — spadek zainteresowania
  - ograniczenia technologiczne w uczeniu sieci o bardziej złożonej architekturze — **głębokie sieci**,
  - sukcesy innych metod uczących się — maszyny oparte na jądrze oraz modele graficzne.

Trzecia fala (od 2006)

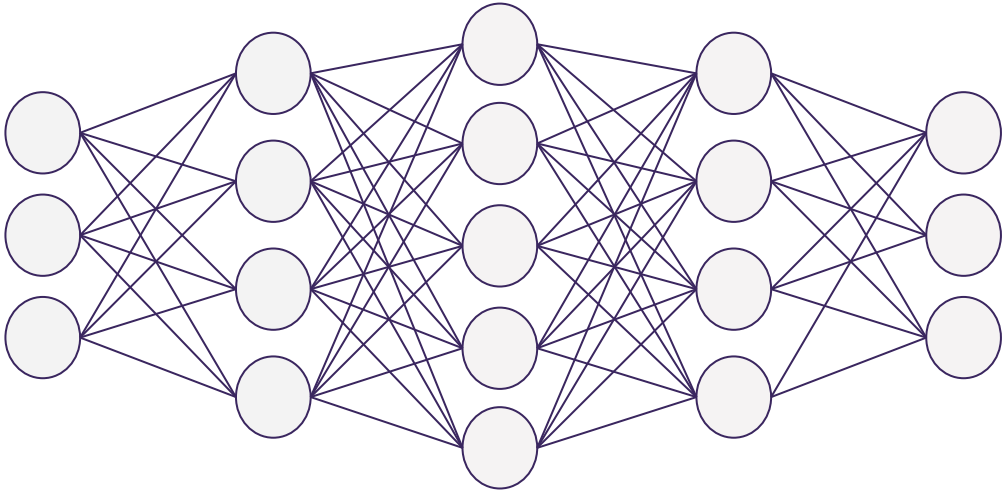
- w 2006 Geffrey Hinton zaproponował sieć neuronową, która mogą być skutecznie uczona za pomocą strategii zachłannego wstępnego szkolenia opartego na warstwach,
- skuteczne metody uczenia sieci neuronowych (w tym wykorzystanie GPU) przyczyniły się do powrotu ich popularności:
  - wykorzystanie dużych zbiorów danych,
  - bardziej złożone architektury sieci,
- trzecia fala — **głębokie sieci neuronowe**.

Z czego wynika „głębokość” sieci?

Głębokość sieci

Liczba warstw ukrytych

Liczba neuronów w warstwach

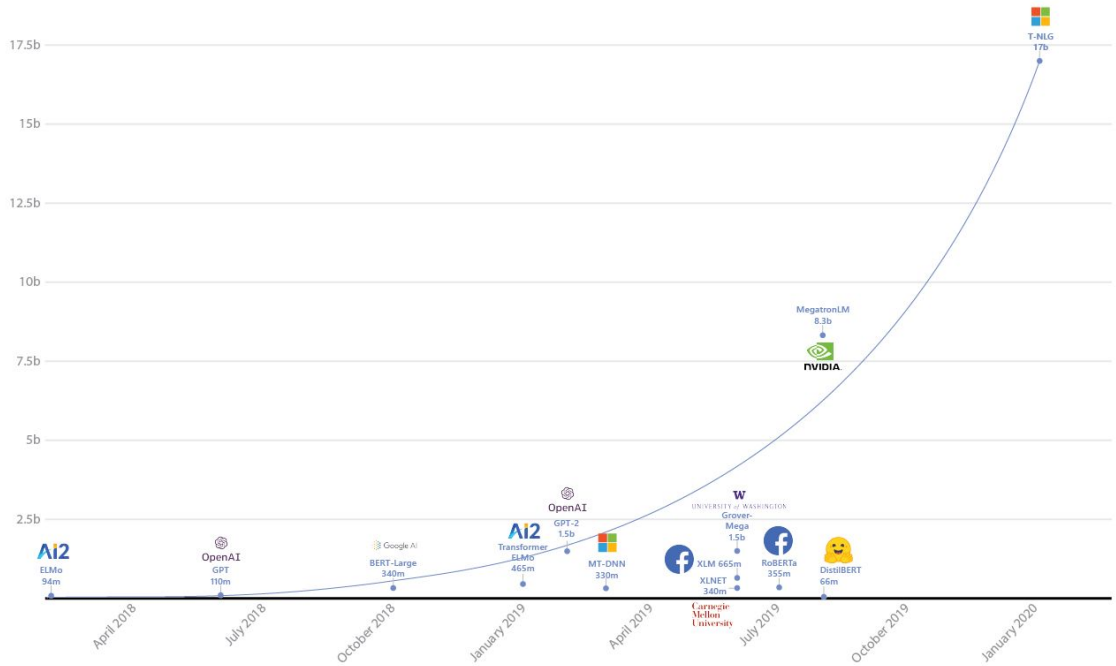


Liczba parametrów sieci

Przykłady głębokich sieci neuronowych

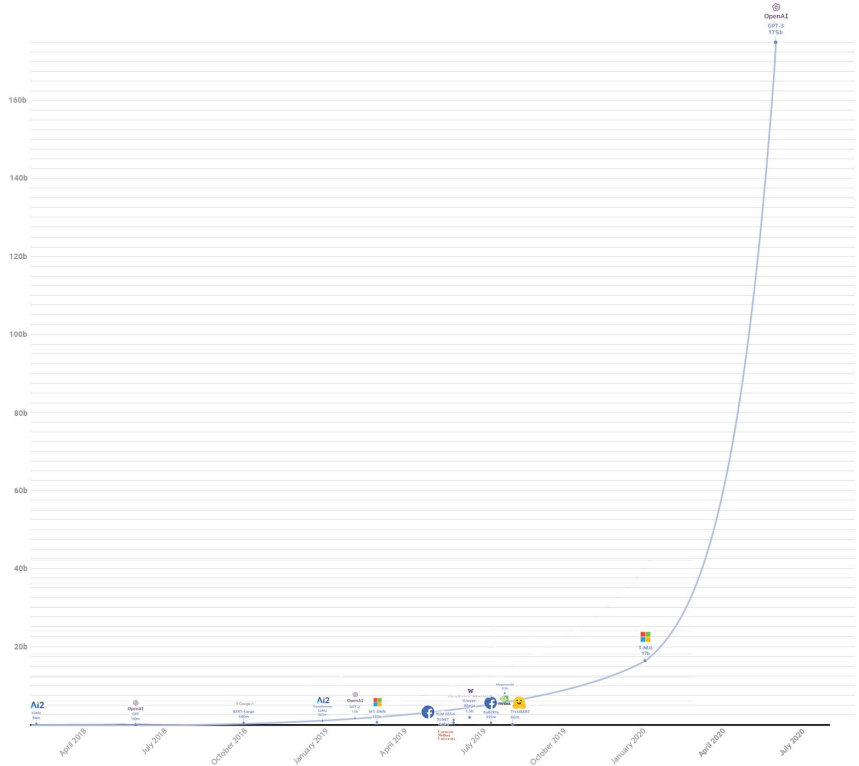
	Model	Liczba parametrów sieci
• 2013	word2vec	3 miliony (300, słownik 10k słów)
• 2018, luty	ELMo	94 milionów
• 2018, październik	BERT base	110 milionów
	BERT large	340 milionów
• 2019, luty	GPT-2	1,5 miliarda
• 2020, czerwiec	GPT-3	175 miliardów

Trend wielkości sieci neuronowych



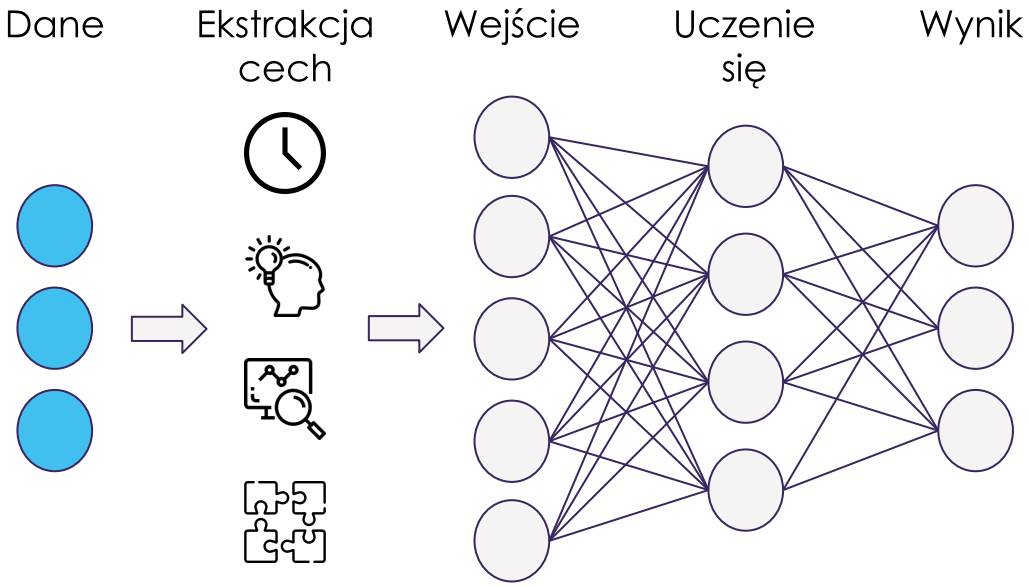
https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

Trend wielkości sieci neuronowych

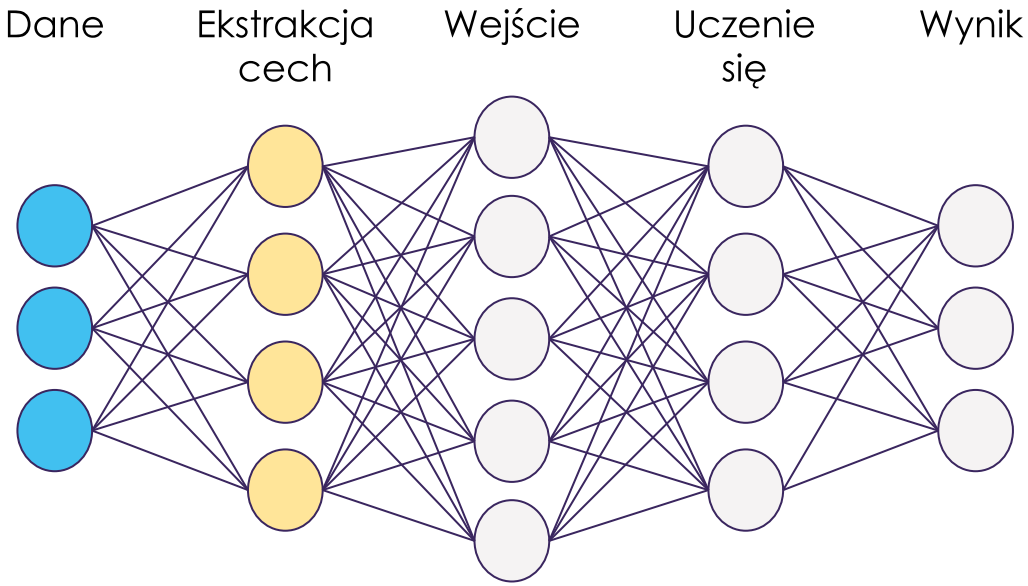


https://bmk.sh/2020/05/29/GPT-3-A-Brief-Summary/

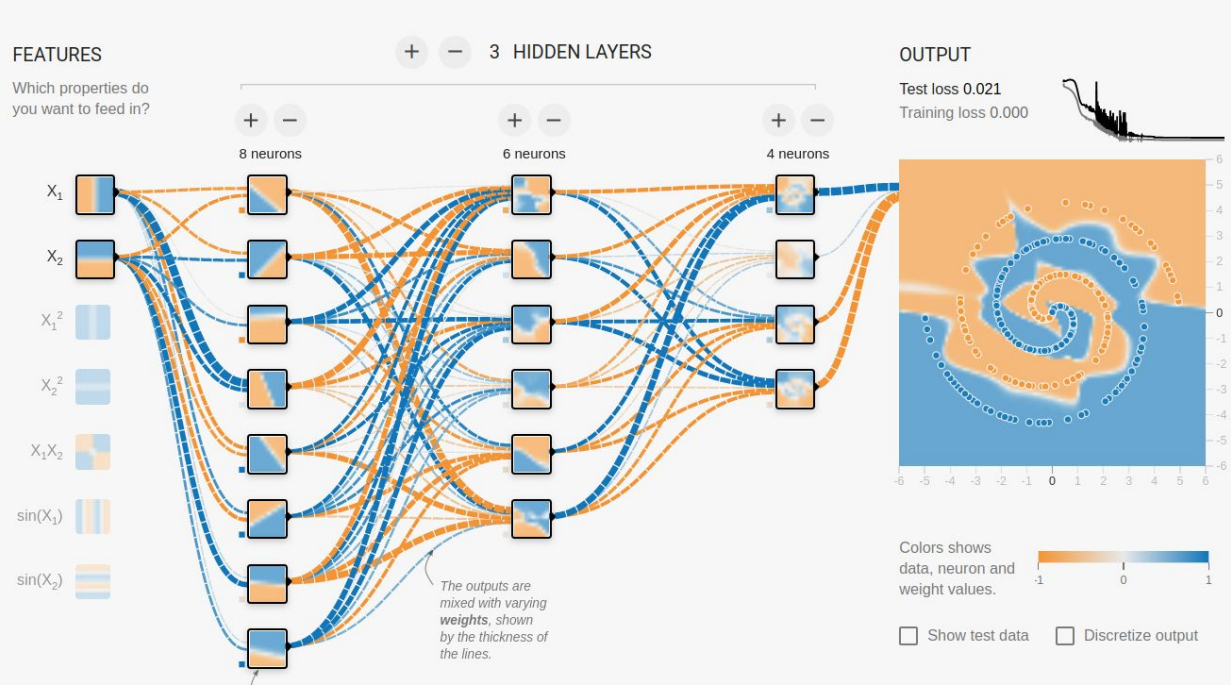
„Tradycyjne” uczenie się sieci



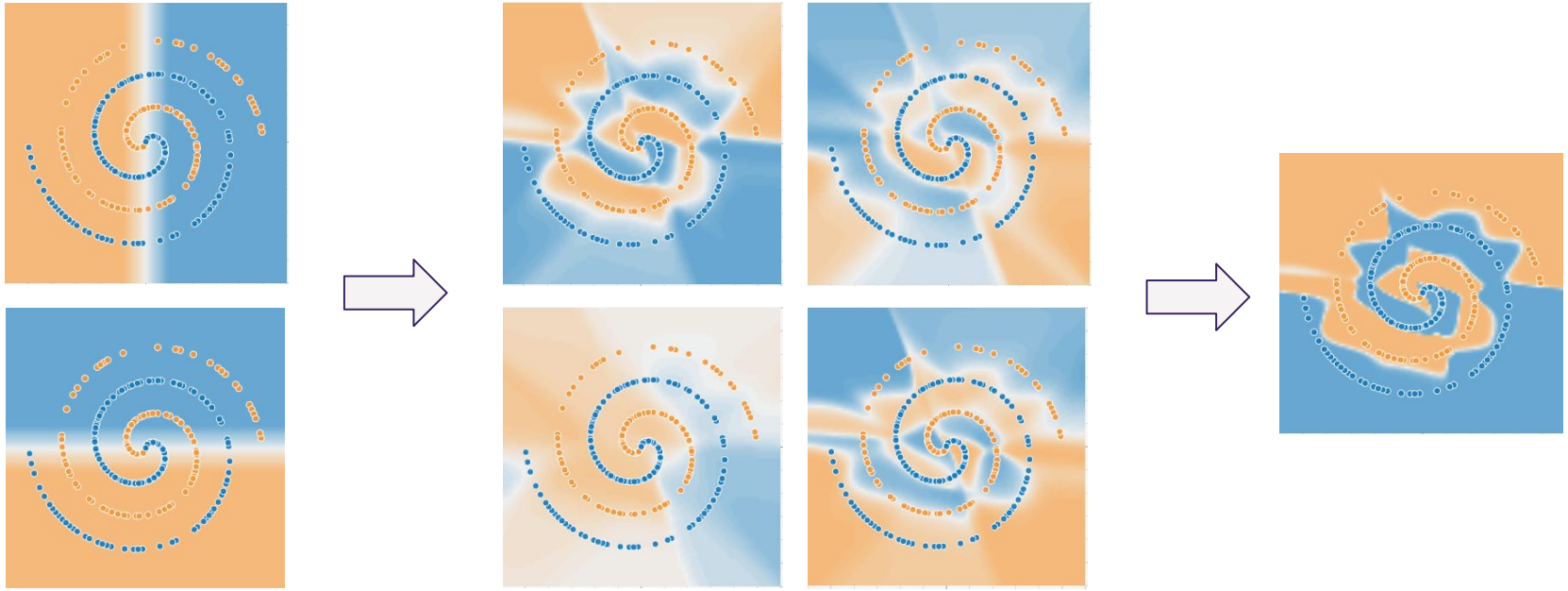
Głębokie sieci neuronowe



Warstwy ukryte jako cechy

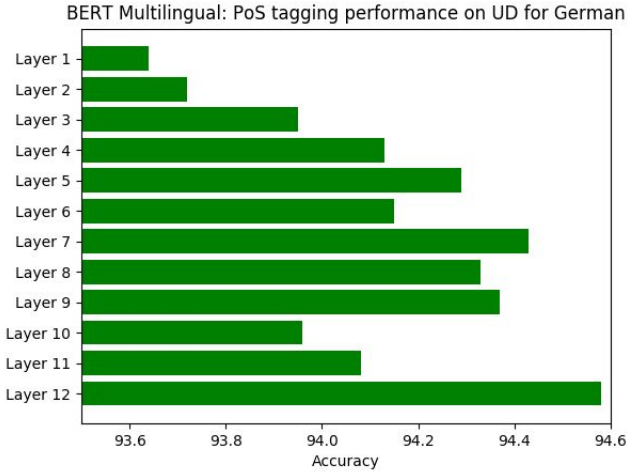


Warstwy ukryte jako cechy



BERT jako model języka

- **Model BERT** był uczony **przewidywać brakujące słowo** w sekwencji.
  „Ala ma ____ kota.”
- 4 x GPU — czas uczenia ok. **34 dni**
- **Warstwy ukryte** BERT-a zostały z powodzeniem wykorzystane jako reprezentacja danych w innych zadaniach NLP, np. tagowanie, NER, itd.
- Uczenie nowego modelu(**fine-tuning**) wykorzystującego reprezentacje z BERT-a jest dużo szybszy i na pojedynczym GPU zajmuje od kilku do kilkudziesięciu minut.

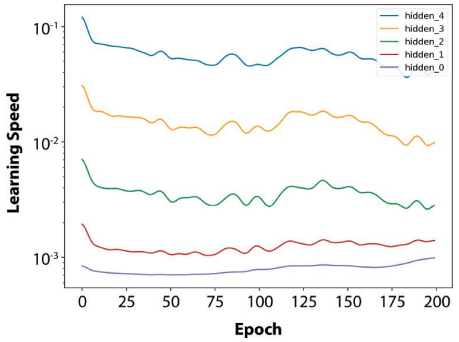


https://github.com/stefan-it/flair-experiments/tree/master/ud-german

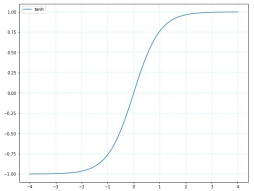
Problemy związane z uczeniem głębokich sieci

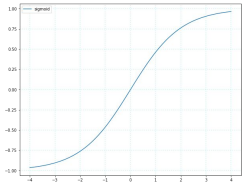
Zanikający gradient

- strata (**loss**) propagowana z **warstwy wyjściowej** w górę sieci w miarę przechodzenia przez kolejne warstwy stopniowo zanika,
- początkowo wagi ustawione są na **losowe wartości**,
- aby uczenie sieci było skuteczne, to wagi pierwszej warstwy muszą możliwie szybko zbliżyć się do optymalnych wartości,
- **funkcje ograniczone** (f. sigmoidalna, tanges) **nasycają się i ich gradient jest niezerowy** w wąskim przedziale aktywacji (blisko zera).



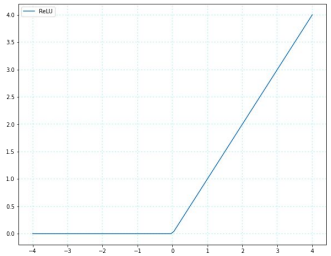
Aglaé Bassens and Grant Beyleveld (2019). Training Deep Networks

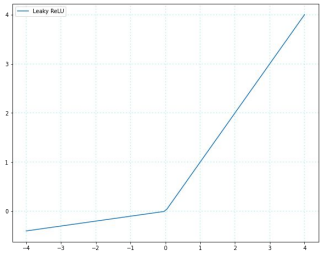


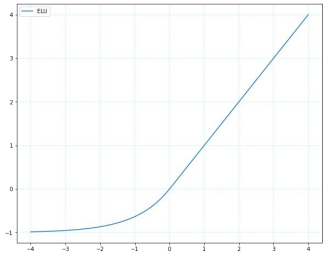


ReLU — Rectified linear unit

- zalety
  - dla aktywnego neuronu gradient nie zanika, co rozwiązuje problem zanikającego gradientu,
  - fragmentami liniowa, dzięki czemu optymalizacja metodami gradientowymi jest skuteczniejsza,
  - efektywna obliczeniowo,
- wady
  - problem umierających neuronów — jak neuron znajdzie się w stanie “0”, to nie jest możliwa jego ponowna reaktywacja (gradient wyniesie 0).
- LeakyReLU, ELU — potencjalne alternatywy dla ReLU.



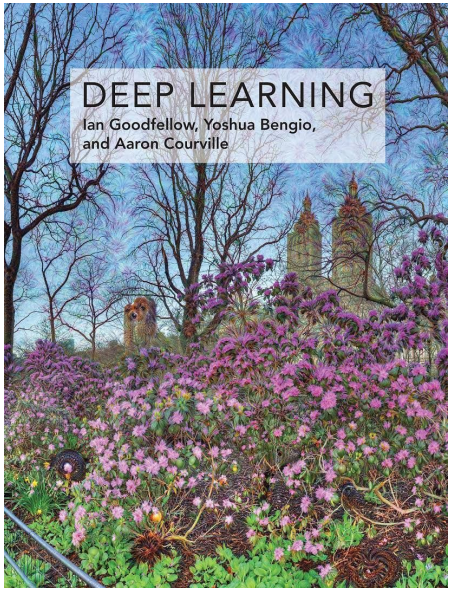




Przeuczenie

- **duża liczba parametrów** (każda waga neuronu to parametr) sprzyja przetrenowaniu — **dostosowanie do danych uczących** i zbyt mała generalizacja modelu,
- sposoby radzenia sobie z problemem przeuczenia sieci:
  - **zbiór walidacyjny** — zatrzymanie procesu uczenia, gdy błąd na zbiorze walidacyjnym zacznie rosnąć,
  - **regularyzacja** (L_1, L_2) — modyfikacja funkcji kosztu w taki sposób, aby wagi były uczone w bardziej uporządkowany sposób, m.in. poprzez wymuszenie wyszukiwania mniejszych wag,
  - **dropout** — tymczasowe usunięcie losowo wybranych neuronów podczas uczenia się sieci. W kolejnych iteracjach usuwany jest inny losowy zestaw neuronów.

Dalsza lektura



- Goodfellow Ian, Bengio Yoshua, Courville Aaron. **Deep learning**

- darmowa książka on-line: http://neuralnetworksanddeeplearning.com/