ardigen
BIOINFORMATICS

Sieci neuronowe

Dawid Rymarczyk, Data Scientist I @ Ardigen

05.12.2017 Kraków, Eksploracja Danych

ardigen
BIOINFORMATICS

https://goo.gl/forms/0fg0aSwemTGhJSSK2

Prezentacja: https://github.com/ddjrym/ED_DL_DR

Po pobraniu **pip install -r req.txt**

Ardigen

- Czy chcesz by rak był chorobą przewlekłą?
- Czy wiesz, że 75% terapii onkologicznych jest nieskutecznych dla większości pacjentów?

CODE AGAINST CANCER

TRUST | KNOWLEDGE | PERSISTENCE | QUALITY

Agenda

- Uczenie maszynowe,
- Sieci neuronowe:
  - wstęp i analiza sieci na przykładzie,
  - algorytm back propagation,
  - funkcje kosztu oraz funkcje aktywacyjne,
  - regularyzacja w sieci. Dropout,
  - sieci splotowe (CNN),
  - sieci rekurencyjne (RNN).

Środowisko pracy:

- Python 3
- Jupyter Notebook
- SciPy pack
- Keras oparty o TensorFlow,
- PyTorch,
- Dane:
  - czasem sztucznie generowane :/
  - ceny nieruchomości,
  - NotMNiST,
  - nagłówki artykułów.

Uczenie maszynowe

Z czego sie uczyć?

CEL

Należy go zdefiniować!

CEL DANE

DANE

- Wyciek danych (data leakage)
- Zła reprezentacja
- Niekompletność

Wyciek danych

- Badanie telemetryczności kierowców przez ubezpieczyciela

Zła reprezentacja

- Z genomu ku priobiotyczności bakterii

CEL
DANE
ALGORYTM

Czy algorytm jest czarną skrzynką?

Przykładowe algorytmy

Almeida–Pineda recurrent backpropagation
ALOPEX
Almeida–Pineda recurrent backpropagation
Backpropagation
Bootstrap aggregating
CN2 algorithm
Constructing skill trees
Dehaene–Changeux model
Diffusion map
Dominance-based rough set approach
Dynamic time warping
Error-driven learning
Evolutionary multimodal optimization
Expectation–maximization algorithm
FastICA
Forward-backward algorithm
GeneRank
Genetic Algorithm for Rule Set Production
Growing self-organizing map
HEXID
Hyper basis function network
IDistance
K-nearest neighbors algorithm
Kernel methods for vector output
Kernel principal component analysis
Leabra
Linde–Buzo–Gray algorithm
Local outlier factor
Logic learning machine
LogitBoost
Manifold alignment
Minimum redundancy feature selection
Mixture of experts
Multiple kernel learning
Non-negative matrix factorization
Online machine learning
Out-of-bag error
Prefrontal cortex basal ganglia working memory
PVLV
Q-learning
Quadratic unconstrained binary optimization
Query level feature
QuickProp
Radial basis function network
Randomized weighted majority algorithm
Reinforcement learning
Repeated incremental pruning to produce error reduction (RIPPER)
Ripple
Rule-based machine learning
Skill chaining
Sparse PCA
State-action-reward-state-action
Stochastic gradient descent
Structured kNN
T-distributed stochastic neighbor embedding
Temporal difference learning
Wake-sleep algorithm
Weighted majority algorithm (machine learning)
K-nearest neighbors algorithm (KNN)
Learning vector quantization (LVQ)
Self-organizing map (SOM)
Logistic regression
Ordinary least squares regression (OLSR)
Linear regression
Stepwise regression
Multivariate adaptive regression splines (MARS)
Ridge regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic net
Least-angle regression (LARS)
Probabilistic classifier
Naive Bayes classifier
Binary classifier
Linear classifier
Hierarchical classifier
Canonical correlation analysis (CCA)
Factor analysis
Feature extraction
Feature selection
Independent components analysis (ICA)
Linear discriminant analysis (LDA)
Multidimensional scaling (MDS)
Non-negative matrix factorization (NMF)
Partial least squares regression (PLSR)
Principal component analysis (PCA)
Principal component regression (PCR)
Projection pursuit
Sammon mapping
t-distributed stochastic neighbor embedding (t-SNE)
AdaBoost
Boosting (machine learning)Boosting
Bootstrap aggregating (Bagging)
Ensemble averaging (machine learning)Ensemble averaging process of creating multiple models and combining them to produce a desired output, as opposed to creating just one model. Frequently an ensemble of models performs better than any individual model, because the various errors of the models "average out."
Gradient boosted decision tree (GBRT)
Gradient boosting machine (GBM)
Random Forest
Stacked Generalization (blending)
Q-learning
State-action-reward-state-action (SARSA)
Temporal difference learning (TD)
Learning Automata
ACDE
Artificial neural network
Association rule learning algorithms
Case-based reasoning
Gaussian process regression
Gene expression programming
Group method of data handling (GMDH)
Inductive logic programming
Instance-based learning
Lazy learning
Learning Automata
Learning Vector Quantization
Logistic Model Tree
Minimum message length (decision trees, decision graphs, etc.)
Probably approximately correct learning (PAC) learning
Ripple down rules, a knowledge acquisition methodology
Symbolic machine learning algorithms
Support vector machines
Random forestRandom Forests
Ensembles of classifiers
Ordinal classification
Information Fuzzy NetworksInformation fuzzy networks (IFN)
Conditional Random Field
ANOVA
Quadratic classifiers
Nearest neighbor (pattern recognition)k-nearest neighbor
Boosting (machine learning)Boosting
Bayesian networks
Hidden Markov models
Bayesian knowledge base
Naive Bayes
Gaussian Naive Bayes
Multinomial Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Bayesian Network (BN)

A więc zbudujmy algorytm!

Zmiksuj:
Własności zbioru danych
Funkcję kosztu
Procedurę optymalizacji
Model

Budując zrozumiesz!

Jupyter Notebook:
Algorytm

CEL
DANE
ALGORYTM

ZAUFANIE I INTERPRETOWALNOŚĆ

"Why should I trust you? Explaining the Predictions of Any Classifier"
YT: https://www.youtube.com/watch?v=KP7-JtFMLo4
Artykuł: http://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf

Sieci neuronowe

Czy Statistical Learning nie wystarczy?

No Free Lunch Theorem for Machine Learning

(Wolpert, 1996)

Motywacja

Przekleństwo wymiarowości

Motywacja

Założenie "klasycznych" algorytmów, że funkcja jest stała na małym przedziale. (Smoothness regularization or local constancy)

Motywacja

Manifold Learning

Czy warto uczyć się sieci neuronowych?

Czy warto uczyć się sieci neuronowych?

Czym jest neuron?

I jak działa?

Czym jest sztuczny neuron?

input_i to zmienna niezależna

Czym jest sztuczny neuron?

Czym jest sztuczny neuron?

Czym jest sztuczny neuron?

Czym jest sztuczny neuron?

Czym jest sztuczny neuron?

Czym jest sieć neuronowa?

Dane wejściowe to też neurony!
Warstwa wejścia

Ostatnia warstwa neuronów to warstwa wyjściowa

Warstwy pomiędzy warstwą wejściową a wyjściową, to warstwy ukryte

Jak działa sieć?

Jupyter Notebook: ANN

Skąd wiemy, że na obrazku jest kotek?

Skąd wiemy, że na obrazku jest kotek?

Skąd wiemy, że na obrazku jest kotek?

Jak sieć się uczy? Gradienty!

Uczenie w oparciu o gradient

Załóżmy, że:
$y = f(x)$, gdzie x,y są rzeczywiste
$f'(x)$ jest pochodną $f(x)$, która mówi jak zmiana na wejściu, wpłynie na zmianę na wyjściu
$f(x+e) \sim= f(x) + ef'(x)$

Jak sieć się uczy?

Funkcja kosztu

Model parametryczny $p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\Theta})$
Entropia wzajemna (cross entropy)

Dlaczego entropia wzajemna a nie błąd średnio-kwadratowy (MSE) lub średni błąd absolutny (MAE)?

Minibatch stochastic method

Jak sieć się uczy?

Funkcja kosztu

Całość procesu uczenia zależy nie tylko od funkcji kosztu, ale również i funkcji aktywacji.

Funkcje aktywacji

Czym jest sztuczny neuron?

Step function

Step function

Pochodna: 0, poza x=0
Zakres wartości: {0, 1}
Zastosowanie: klasyfikacja binarna lub grupowanie

Czy jest większy sens jej używać?

Funkcja liniowa

Funkcja liniowa

Pochodna: a, jeżeli y=ax
Zakres wartości: (-inf, inf)
Wartość proporcjonalna do wejścia

Czy jest możliwa nieliniowość przy takiej funkcji aktywacji?

Sigmoid

Sigmoid

Zakres wartości: (0, 1) **LUBIMY GO!**

Dobra do klasyfikacji

Vanishing gradient

Pochodna:

Tangens hiperboliczny

Tangens hiperboliczny

tanh(x) = 2sigmoid(2x) - 1
Vanishing gradient
Zakres wartości: (-1;1)

Rectified Linear Unit

ReLU

Nieliniowa
Każda funkcja może być aproksymowana za pomocą wielu funkcji ReLU
Rzadka aktywacja
Martwe neurony
Zakres wartości: (0;inf)
Wiele wariacji, np. Leaky ReLU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$

Exponential Linear Unit

ELU

Wszystkie zalety ReLU

Neurony nie umierają!

Softmax

Softmax

Najczęściej używane jako warstwa wyjściowa przy klasyfikacji.

Backpropagation

Architektura sieci

O czym myśleć przy architekturze?

Głębokość
Szerokość warstw

Universal
Approximation Theorem

Lepiej generalizują się
modele głębokie

Deep Learning and the Information Bottleneck Principle

N. Tishby

E = MC^2

Energy = Milk + Cofee^2

Regularyzacja

Parameter Norm Penalties

$$\widehat{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

Norma L2 (Ridge)

$$\widehat{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

$$\Omega(\theta) = \frac{1}{2}\|w\|_2^2$$

$$\widehat{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \mathrm{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \frac{\alpha}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

$$\nabla_{\mathbf{w}}\widehat{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \alpha\mathbf{w} + \nabla_{\mathbf{w}}\mathrm{J}(\mathbf{w}; \mathbf{X}, \mathbf{y})$$

$$\mathbf{w} \leftarrow (1 - \epsilon\alpha)\mathbf{w} - \epsilon\nabla_{\mathbf{w}}\mathrm{J}(\mathbf{w}; \mathbf{X}, \mathbf{y})$$

w_2
Minimize cost
$\lambda \|\mathbf{w}\|_2^2$
w_1
Minimize penalty
Minimize cost + penalty

Norma L1 (taka jak w Lasso)

$$\widehat{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

$$\Omega(\theta) = \|\mathbf{w}\|_1 = \sum_i |w_i|$$

$$\widehat{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \alpha\|\mathbf{w}\|_1 + J(\mathbf{w}; \mathbf{X}, \mathbf{y})$$

$$\nabla_w \widehat{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \alpha \mathrm{sign}(\mathbf{w}) + \nabla_w J(\mathbf{w}; \mathbf{X}, \mathbf{y})$$

w_2
$\lambda\|\mathbf{w}\|_1$
w_1
Minimize cost + penalty
(w_1 = 0)

L1 vs L2

- dzięki L1 następuje wybór cech
- L2 ma rozwiązanie analityczne, więc łatwiej i szybciej je znaleźć

Możemy dodać warunek do norm!

$$\widehat{J}(\theta; \mathbf{X}, \mathbf{y}) = \alpha(\Omega(\theta) - \mathrm{k}) + \mathrm{J}(\theta; \mathbf{X}, \mathbf{y})$$

$$\mathcal{L}(\theta, \alpha; \mathbf{X}, \mathbf{y}) = \mathrm{J}(\theta; \mathbf{X}, \mathbf{y}) + \alpha(\Omega(\theta) - \mathrm{k})$$
$$\theta^* = \arg\min_{\theta} \max_{\alpha, \alpha \geq 0} \mathcal{L}(\theta, \alpha)$$

$$\theta^* = arg \min_{\theta} \mathcal{L}(\theta, \alpha^*; \mathbf{X}, \mathbf{y}) = \arg\min_{\theta} \mathrm{J}(\theta; \mathbf{X}, \mathbf{y}) + \alpha^* \Omega(\theta)$$

Czasami możemy chcieć explicite ustawić ograniczenie dla regularyzacji. Na przykład mając informację w oparciu o wiedzę domenową problemu.

Reprojekcja.

Regularyzować wagi kolumnami.

Generowanie sztucznych danych

- transformacje
- szum do danych
- zaprojektuj swoją przemyślaną wcześniej transformację

Wykorzystanie szumu nie tylko do danych wejściowych

Dodanie szumu do wag

Wykorzystanie szumu nie tylko do danych wejściowych

Dodanie szumu do y

Semi-supervised

Mamy obserwacje i przypisane do nich etykiety. Modelujemy P(**y** | **x**). Ale nie wszystkie obserwacje mogą posiadać etykiety.

Wielonauczanie

Wczesne zatrzymanie

Early Stopping

Łączenie parametrów i ich współdzielenie

Łączenie za pomocą normy, np: ||w1 - w2||

Współdzielenie między dwoma modelami

Rzadka reprezentacja

OMP-k (Orthogonal Matching Pursuit)

$$arg \min_{h, \|h\|_0 < k} \|x - Wh\|^2$$

Ensembling

Bagging,
Boosting

DROPOUT

- zmniejsza pojemność modelu, aby zapobiec trzeba zwiększyć jego głębokość,
- kiedy mało dany, Bayesian Neural Net,
- powoduje, że neurony są jak geny

Adversarial training

5 minut przerwy polecam ;)

Konwolucyjne (splotowe ;)) sieci neuronowe [CNN]

CNN jak wskazuje nazwa wykorzystują splot

Świetne zastosowanie do danych, które tworzą siatkę. Np.: szeregów czasowych (1D), obrazów (2D)

Sieć nazywamy konwolucyjną, gdy choć jedna z warstw używa splotu zamiast mnożenia macierzy

A więc pamiętamy czym jest splot?

Zyski ze splotu

Filtr o długości 3

Zyski ze splotu

Filtr o długości 3

Zyski ze splotu

Filtr o długości 3

Zyski ze splotu

Rzadkie połączenia!

Zyski ze splotu

Współdzielenie parametrów

Zyski ze splotu

Equivariance

Zyski ze splotu

Equivariance

Typowa warstwa konwolucyjna

Pooling

Uogólnia odpowiedzi w prostokątnym sąsiedztwie

Zyski z Poolingu: NIEZMIENNOŚĆ

MaxPooling

O warstwie splotowej można myśleć jak o prawdopodobieństwie początkowym (prior)

Filozoficznie, a może i nie o CNN

http://scs.ryerson.ca/~aharley/vis/conv/flat.html

A później Jupyter Notebook: NotMNiST

Rekurencyjne sieci neuronowe (RNN)

Neuron ukryty w RNN

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta)$$

$$\mathbf{h}^{(t)} = g^{(t)}(\mathbf{c}^{(t)}, \mathbf{c}^{(t-1)}, \ldots, \mathbf{c}^{(1)}, \mathbf{c}^{(0)},)$$

Współdzielimy parametry

RNN: podstawowe wzorce

Sekwencja na sekwencje

output w każdym kroku

output w każdym kroku, ale rekurencja tylko z outputu

RNN: podstawowe wzorce

Sekwencja na wektor
o zadanej długości

output po przetworzeniu
całej sekwencji

Co się dzieje w RNN?

$$\mathbf{a}^{t} = \mathbf{b} + \mathbf{W}\mathbf{h}^{t-1} + \mathbf{U}\mathbf{x}^{(t)}$$
$$\mathbf{h}^{(t)} = \tanh(\mathbf{a}^{t})$$
$$\mathbf{o}^{(t)} = \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)}$$
$$\hat{\mathbf{y}}^{(t)} = \mathrm{softmax}(\mathbf{o}^{(t)})$$

A jaka będzie wartość funkcji kosztu?

Brak możliwości obliczania na wielu wątkach :'(

Teacher forcing

Motywacja:
Brak backprop, bo brak połączeń między ukrytymi neuronami

Wada: Widzimy tylko y dane do treningu. Problem z generalizacją. Jak sobie z tym poradzić?

Ale jak to modeluje sekwencje?

Ale skąd wiemy jak długa jest sekwencja?

1. W słowniku mamy symbol końca sekwencji.
2. Wprowadzić nowy neuron wyjściowy, który jest binarny.
3. Wprowadzić nowy neuron wyjściowy, który przewiduje długość sekwencji. Wymaga też nowego neuronu wejściowego.

A czy możemy z wektora dostać sekwencję?

Architektura enkoder - dekoder

- Wprowadzamy kontekst C
- kontekst musi być odpowiednio długi

Leaky unit

1. Pomagają poradzić sobie z długofalowymi zależnościami
2. Są to neurony z liniowy połączeniem do samych siebie z wagą bliską 1.
3. Zapewniają, że w przyszłości, później, później neuron może do nich sięgnąć.

Long Short-Term Memomory (LSTM)

- Neurony typu leaky są bardzo niebezpieczne, dlatego przechodzimy na bramki

Notebook RNN