Convolutional Neural Networks in High Energy Physics

Adam Novotný

Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague

June 23, 2019

Overview

1. Basic concepts of neural networks
2. Convolutional neural networks
3. protoDUNE experiment results

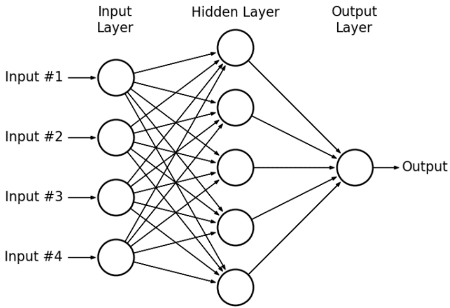
Basic concepts of neural networks

- supervised learning algorithm (imitation of biological neural network),
- let $\mathcal{T} = ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(p)}, y^{(p)}))$, where $p \in \mathbb{N}$ is number of observations, $j \in \widehat{p} : \boldsymbol{x}^{(j)}$ is input vector and $j \in \widehat{p} : y^{(j)}$ is labeled output, be a dataset,
- $\mathcal{T}$ is further divided into training set, testing set and validation set (different techniques like k-fold cross-validation can be used),
- concept of neural networks is built upon universal approximation theorem

Most widely used architecture

- multilayer perceptron (MLP)

Figure: Multilayer perceptron with 4 inputs, 1 output and 1 hidden layer.



$$\text{in each layer for each neuron} : z = f\left(\sum_{i=1}^{|\mathcal{I}|} w_i x_i + \theta\right) \quad (1)$$

Optimization methods

- *loss function* measures the difference between output $\widehat{y}^{(j)}$ produced by neural network (with input $\boldsymbol{x}^{(j)}$) and labeled $y^{(j)}$ from $\mathcal{T}$,
  - regression: mean squared error as $L = \frac{1}{p}\sum_{j=1}^{p}\left(y^{(j)} - \widehat{y}^{(j)}\right)^2$,
  - classification:
    cross-entropy for binary classification as:

$$L = -\frac{1}{p}\sum_{j=1}^{p} y^{(j)}\log(\widehat{y}^{(j)}) + (1 - y^{(j)})\log(1 - \widehat{y}^{(j)}),$$

    cross-entropy for multi-class classification as:

$$L = -\frac{1}{p}\sum_{j=1}^{p}\sum_{i=1}^{\#\text{classes}} y_i^{(j)}\log(\widehat{y}_i^{(j)}),$$

  - categorical cross-entropy = softmax (on $\widehat{y}^{(j)}$) + cross-entropy,
- goal is to find global minimum of L with respects to weights matrix W

Optimization & Regularization methods

- gradient descent,
- stochastic gradient descent (randomly selected samples to compute gradients) and its numerous modifications, e. g. Adam (Adaptive Moment Estimation; adaptive change of step length etc.),
- backpropagation algorithm,
- *regularization* to loss function:
  - to loss function: e. g. $L = \underbrace{\frac{1}{p}\sum_{i=1}^{p} L_j}_{\text{data loss}} + \underbrace{\lambda \sum_{k}\sum_{l} W_{k,l}^2}_{\text{regularization loss}}$ (L_2 norm penalty),
  - in architecture: dropout (randomly excluding neurons)

Measures of classification goodness

- accuracy (simple correct predictions:all predictions on test set ratio)
- confusion matrix

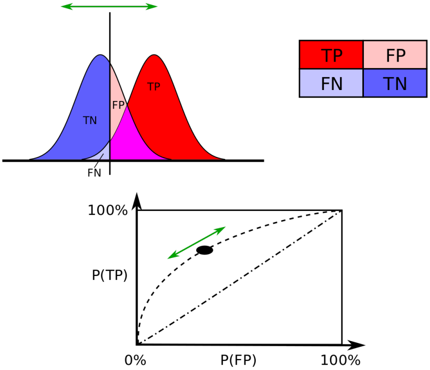
Figure: Confusion matrix definition.

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment): positives	Actual Value (as confirmed by experiment): negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

- F_1 score for binary classification, equals $2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$, where Precision $= \frac{\text{TP}}{\text{TP}+\text{FP}}$ and Recall $=$ True Positive Rate (TPR) $= \frac{\text{TP}}{\text{TP}+\text{FN}}$.

- receiver operating characteristic (ROC), area under curve (AUC)

Figure: ROC and AUC vizualization.

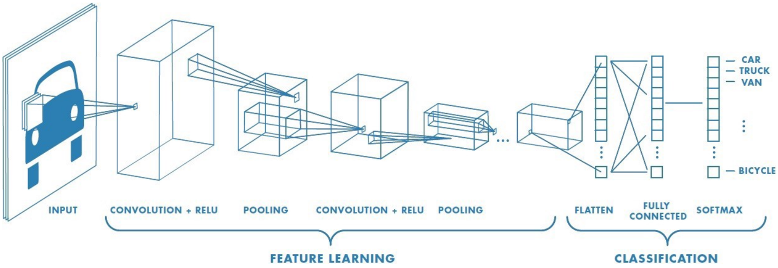


True Positive Rate (TPR) $= \frac{TP}{TP+FN}$, False Positive Rate (FPR) $= \frac{FP}{TN+FP}$

Convolutional neural network

- specially effective with visual data, widely used in computer vision
- layers typical for CNN: convolution and pooling layers (and flatten)
- consists of feature learning and classification

Figure: Demonstration of CNN architecture.



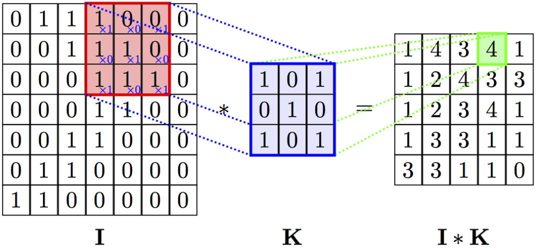
Convolution layer

- 2D convolution with kernel K

$$(I * K)(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} I(m,n)K(i-m, j-n)$$

- stride corresponds to pixel shift
- padding (e. g. zero padding) used for not lowering dimension (information loss prevention)

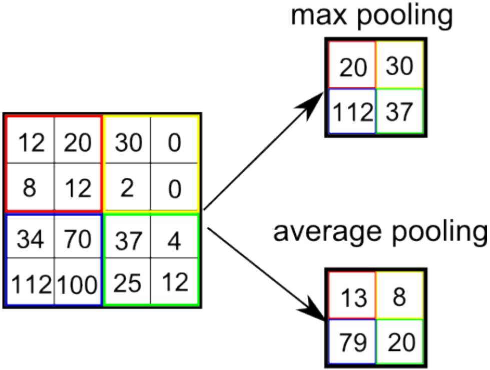
Figure: Convolution visualization (with stride 1).



Pooling

- method for dimensionality reduction
- max pooling (most used), average pooling

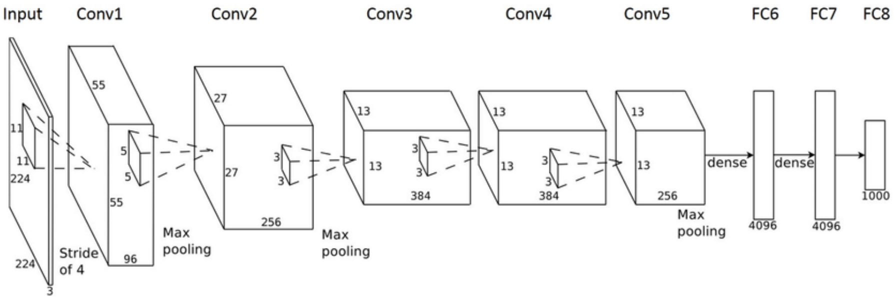
Figure: Pooling.



Convolutional neural network architectures

- kernels in the first convolutional layer detect low-level features (edges and curves),
- kernels in higher layers encode more abstract features,
- by stacking several convolutional and pooling layers, higher-level feature representations could be gradually extracted,

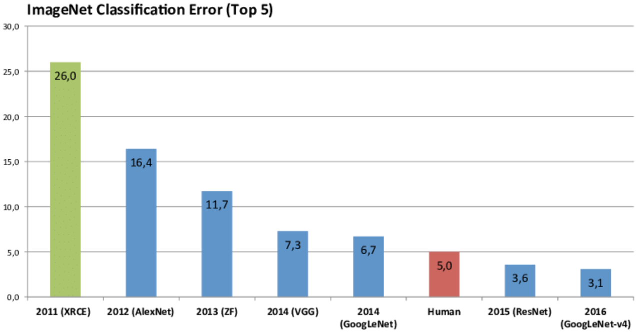
Figure: Example of CNN architecture (AlexNET).



Convolutional neural network architectures

- benchmark datasets: MNIST, cifar10, ImageNet (ILSVRC challenge)
- breakthrough in 2012 (AlexNet)
- great successes in next years: LeNet (Inception), VGG, ResNet, ResNext; more in Mira's presentation

Figure: Different CNN architectures.



Experiment protoDUNE

- Liquid Argon Time Projection Chambers (LArTPC) experiment
- DUNE, protoDUNE: Fermilab, CERN collaboration
- dataset primarily focused on classification of electron, muon and tau neutrino; secondarily on other variables' classification (interaction type, number of particles in the process)
- optimizer: SGD
- binary cross-entropy (for binary classification of neutrino/antineutrino) and categorical cross-entropy (for multi-class classification for others)
- architecture: ResNet18 with primarily with one output layer, secondarily with seven output layers
- in Tensorflow 1.13.1, Python 3.6

Experiment protoDUNE

Overall results by DUNE CNN team (accuracies in %) tested on protoDUNE Monte Carlo samples (MCC8.1):

- neutrino/antineutrino: 73.5,
- flavour: 90.3,
- interaction type: 71.5,
- # protons: 81.2 (0, 1, 2, 3+),
- # pions: 84.1 (0, 1, 2, 3+),
- # pizeros: 90.9 (0, 1, 2, 3+),
- # neutrons: 99.1 (0, 1, 2, 3+; but almost all the events have 0 neutrons)

Future goals & Conclusion

Future goals:

- other metrics
- 3D convolutional neural network
- sparse convolutional network

Conclusion:

- we have given a brief introduction into neural network and its convolution aspects
- we have discussed protoDUNE experiment classification results

Thank you for your attention

Variables' classes

- neutrino/antineutrino
- flavour: CC (charged current) ν_μ, CC ν_e, CC ν_τ, NC (neutral current),
- interaction type: CC quasi electric, CC Res, CC DIS, CC other
- # protons: 0, 1, 2, 3+,
- # pions: 0, 1, 2, 3+,
- # pizeros: 0, 1, 2, 3+,
- # neutrons: 0, 1, 2, 3+; but almost all the events have 0 neutrons