Deep Learning

Movie genre classification

MORFEUSZ

1 Dataset

In the project, a database entitled *Movie Genre from its Poster* has been used. The database is provided by kaggle, an online machine learning and data science community. Figure 1 shows exemplary posters from the studied database.

Figure 1: Exemplary posters.

The database include 40109 records, containing information about films' IMDB IDs, website links, titles, IMDB scores and genres. For each entry, a link to a corresponding poster is provided. Table 1 displays an exemplary record of a film. In the analysis, poster graphics along with genre information have been used, in order to perform film genre classification.

Table 1: An exemplary record from the poster database.

imdbId	120737
Imdb Link	http://www.imdb.com/title/tt120737
Title	The Lord of the Rings: The Fellowship of the Ring (2001)
IMDB Score	8.8
Genre	Adventure|Drama|Fantasy
Poster	"https://images-na.ssl-images-amazon.com/images/M/MV5BNmFmZDdkOD MtNzUyMy00NzhhLWFjZmEtMGMzYjNhMDA1NTBkXkEyXkFqcGdeQXV yNDUyOTg3Njg@._V1_UY268_CR0,0,182,268_AL_.jpg"

In total, the dataset contains 28 genres: Action, Adult, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Game-Show, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, Talk-Show, Thriller, War, Western. For the neural network training, five most common occurring classes (Action, Comedy, Drama, Horror, Romance) have been chosen.

2 Deep Learning

2.1 Layers

2.1.1 Dense layer

The base layer of a neural network is a dense layer, also called fully connected layer. In this type of layers, each neuron in a given layer receives input from all neurons of the previous layer.

To use this layer for image recognition, a graphic has to be flatten into a pixel array. In the case of monochrome images, the obtained matrix is in the shape of (N, 1). For RGB graphics, the matrix is in the shape of (N, 3) with three columns for each color channel.

In order to use this layer, the number of neurons has to be specified. In the project a number of 128 is used in a neural network model.

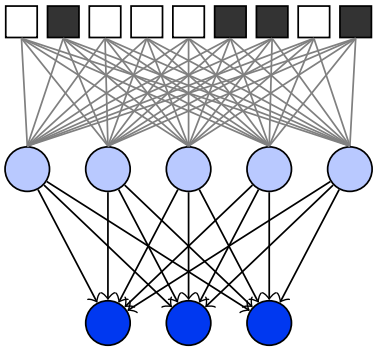


Figure 2: A scheme of a dense layer.

2.1.2 Convolutional layer

Convolutional layers are most commonly applied to analyse images. In this case, the input is a tensor with a shape of (inputs, height, width, channels). This type of layers convolve the input and pass the result to the next layer.

In order to use this layer, number of filters and kernel size have to be chosen. Kernel size parameter is a 2-tuple specifying both width and height of the 2D convolution window. Filters parameter determines the number of kernels used in the layer. Additionally, padding type can be chosen between valid, which allows for the reduction of the spatial dimensions, and same, which results in padding with zeros around the input image.

In the project, 128 filters and (3, 3) kernel size are used. The padding is set to the same value.

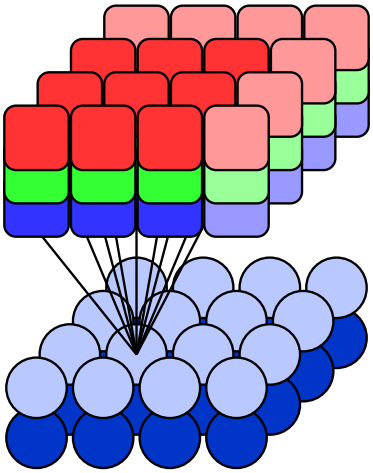


Figure 3: A scheme of a convolutional layer.

2.1.3 Dropout

Dropout, also called dilution, is a method, which effectively reduces overfitting in neural networks. In statistics, overfitting is modeling error that corresponds too closely to a particular set of data, and therefore may fail to predict future observations.

The term dropout refers to randomly dropping out, or omitting, units during the training process. Nodes are either kept with probability p or dropped out with probability 1-p. p parameter has to be specified in the model.

In the project 1-p, parameter is set to 0.25 after each convolutional and 0.5 after each dense layer.

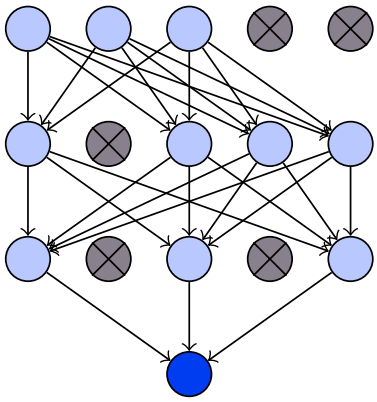


Figure 4: A scheme of a dropout method.

2.2 Activation function

The activation function of a node determines the output of the node given one input or set of inputs. The activation functions are divided into 2 types: linear and non-linear. However, only nonlinear functions allow neural networks to compute nontrivial problems.

2.2.1 Sigmoid

The Sigmoid function is given by the folmula

$$S(x) = \frac{1}{1 + e^{-x}}.$$

The function is monotonic and differentiable. The output of sigmoid function is between 0 and 1. Therefore, it is especially useful for models, where the probability is predicted. In the project, the sigmoid function is used in the last dense layer.

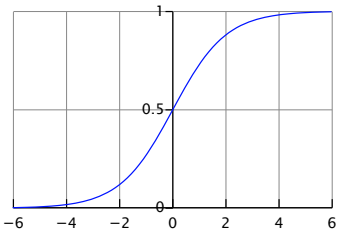


Figure 5: Sigmoid function.

2.2.2 ReLU

The Rectified Linear Unit (ReLU) function is given by the folmula

$$R(x) = \max(0, x).$$

Both the function and its derivative are monotonic. The function ranges from 0 to infinity. ReLU is the most used activation function in the convolutional neural networks. In the project, ReLU function is used in all but the last layer.

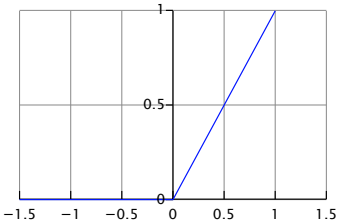


Figure 6: ReLU function.

2.2.3 Softmax

The Softmax function is given by the folmula

$$\sigma(\vec{z})_i = \frac{e^{\vec{z}_i}}{\sum_{j=1}^{K} e^{\vec{z}_j}},$$

where $\vec{z}$ denotes a vector of inputs. The softmax function is a more generalized logistic function and is used for multiclass classification.

2.3 Loss function

The cross-entropy Loss is the most used loss in neural networks. The cross-entropy loss is defined as

$$CE = -\sum_{i}^{C} t_i \log(s_i),$$

where t_i and s_i denote truth and neural network score for i^{th} class in C. An activation function has to be applied to the scores before the cross-entropy loss computation.

2.3.1 Categorical crossentropy loss

Categorical crossentropy loss, also called softmax loss, is used for multi-class classification. This loss trains a neural network to output a probability over the C classes.

2.3.2 Binary crossentropy loss

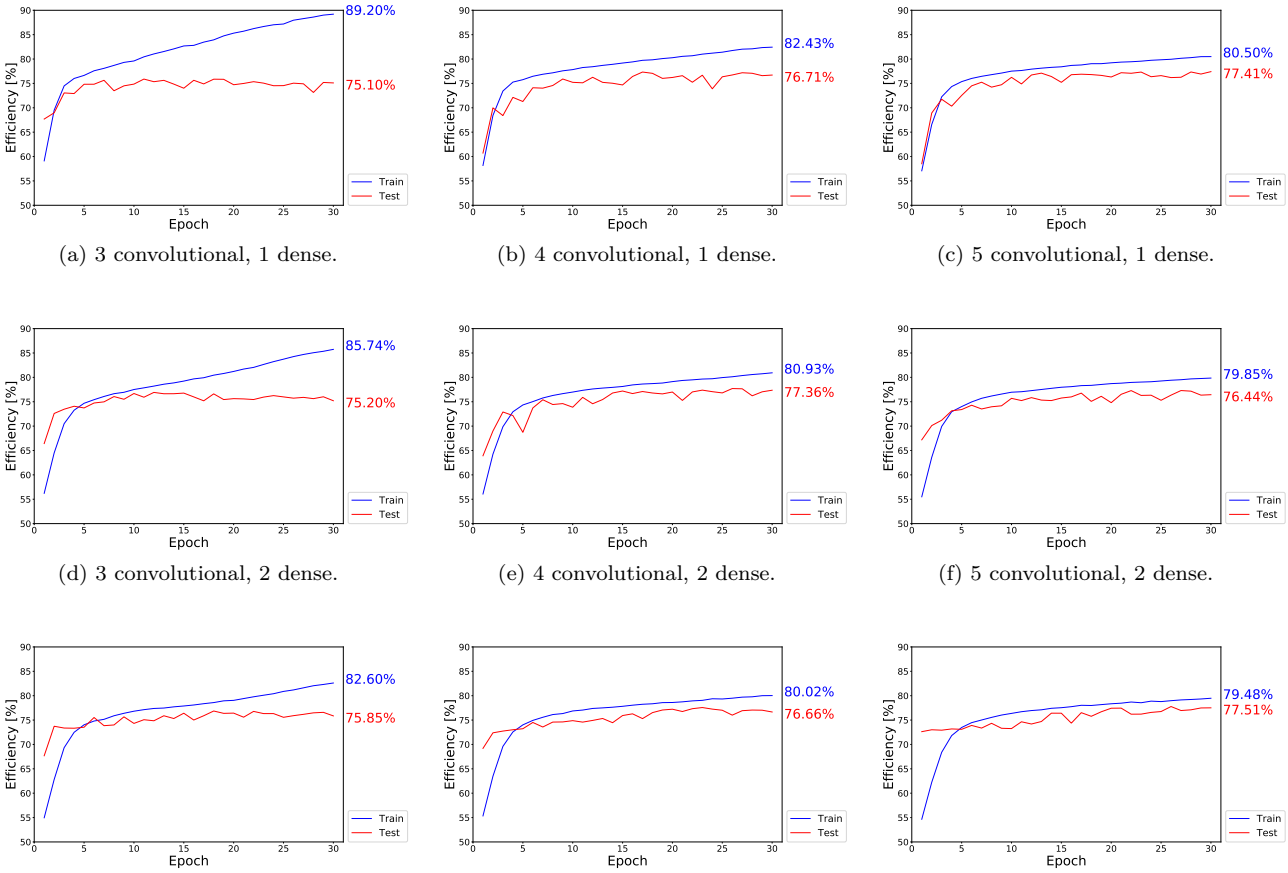
Binary crossentropy loss, also called sigmoid cross-entropy loss, is used for multi-label classification. This loss is independent for each class. In the project, binary crossentropy loss is applied.

3 Results

In the project, 9 convolutional neural networks are studied in total. The models are trained to classify film posters into one or more genres from the following set: Action, Comedy, Drama, Horror, Romance. The combinations of 3,4,5 convolutional layers and 1,2,3 dense layers are compared. All presented models are trained for 30 epochs. For the validation purpose, 10% of the data is allocated to the test dataset.

3.1 Efficiency

Figure 7 shows model efficiency as a function of epochs for the training and test datasets. For networks with the smallest number of convolutional layers, large discrepancies between train and test efficiencies are observed. These differences are caused by model overfitting, which is a well-known effect in neural networks. The model with 5 convolutional and 3 dense layers shows the best performance with test efficiency reaching 77,51%.



(a) 3 convolutional, 1 dense. (b) 4 convolutional, 1 dense. (c) 5 convolutional, 1 dense.

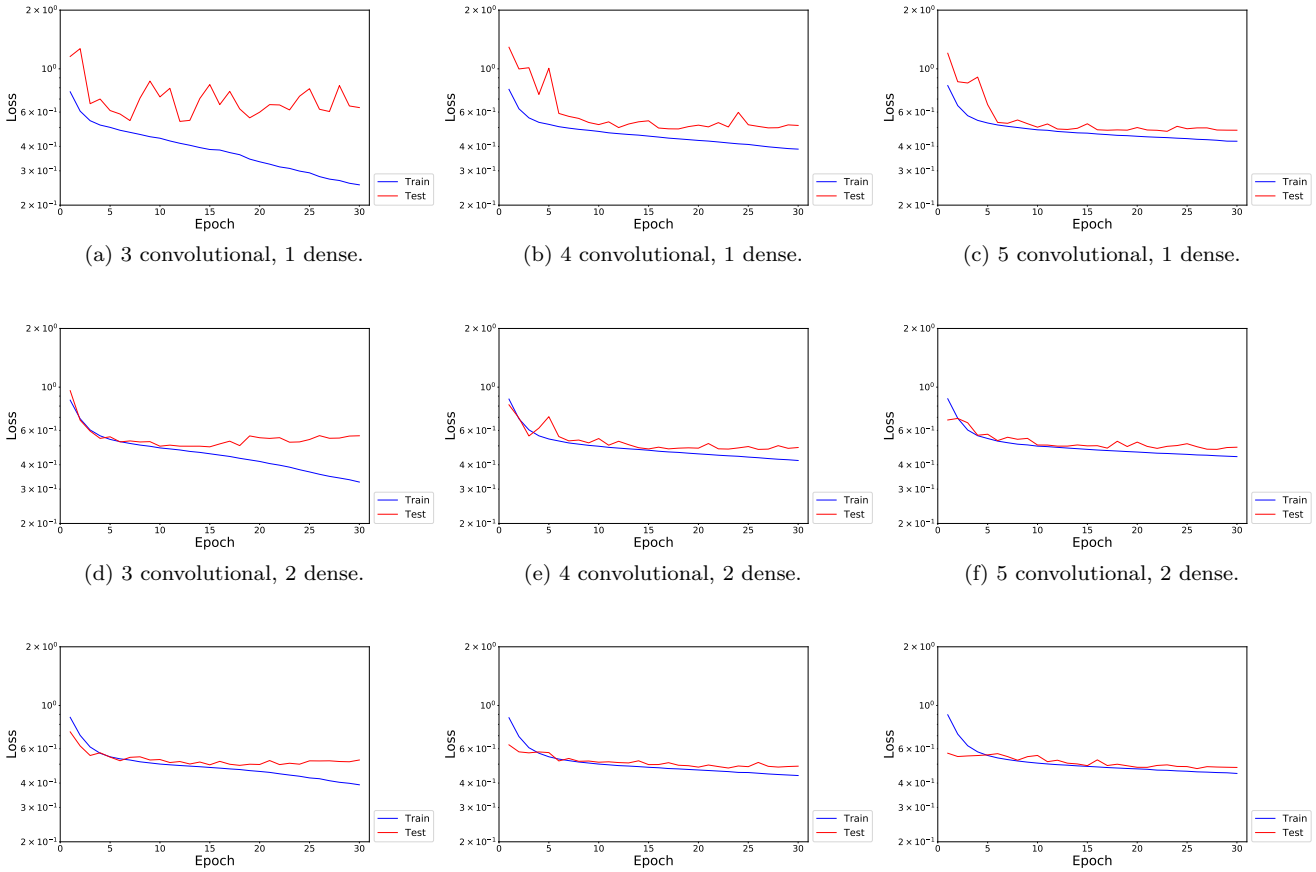
(d) 3 convolutional, 2 dense. (e) 4 convolutional, 2 dense. (f) 5 convolutional, 2 dense.

(g) 3 convolutional, 3 dense. (h) 4 convolutional, 3 dense. (i) 5 convolutional, 3 dense.

Figure 7: Efficiency as a function of epochs.

3.2 Loss

Figure 8 presents loss functions as a function of epochs for the training and test datasets. The network with 4 convolutional and 1 dense layer shows the largest discrepancy between train and test datasets caused by overfitting. Additionally, significant fluctuations in test loss function are observed. The best performance is obtained with 5 convolutional and 3 dense layer model. This network shows the smallest differences between train and test datasets and achieves the lowest test loss value of 0.4812.



(a) 3 convolutional, 1 dense. (b) 4 convolutional, 1 dense. (c) 5 convolutional, 1 dense.

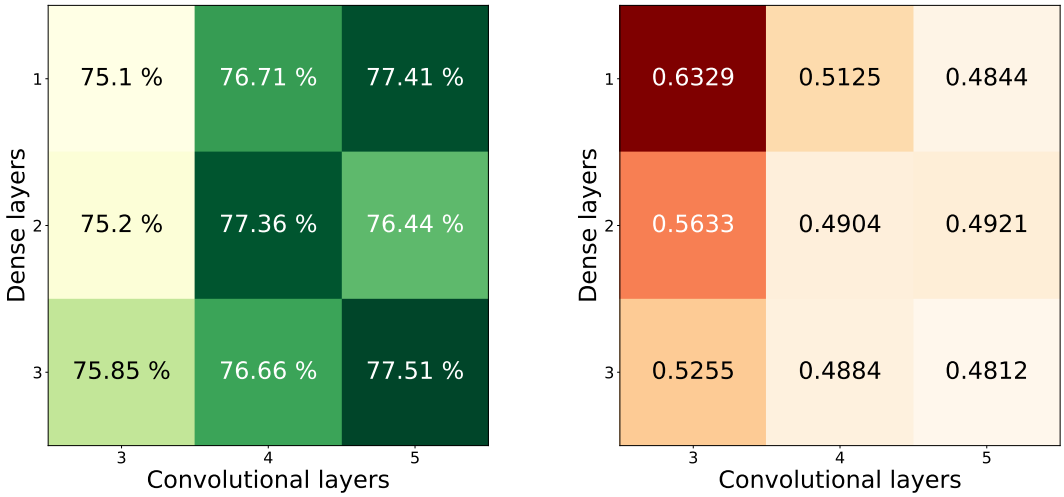
(d) 3 convolutional, 2 dense. (e) 4 convolutional, 2 dense. (f) 5 convolutional, 2 dense.

(g) 3 convolutional, 3 dense. (h) 4 convolutional, 3 dense. (i) 5 convolutional, 3 dense.

Figure 8: Loss function as a function of epochs.

3.3 Model comparison

In Figure 9, efficiency and loss in the last epoch have been compared for all trained models.



(a) Efficiency. (b) Loss.

Figure 9: Comparison of efficiency (left) and loss (right).

The networks with 3 convolutional layers show the worst overall performance. The efficiency raises significantly with the number of convolutional layers in the model. The number of dense layers has much weaker impact on the network performance.

3.4 Examples

Lastly, four examples of genre predictions performed by the model with 5 convolutional and 3 dense layers are presented in Figure 10. The model returns weights ranging from 0 to 1 for each class independently.

Prediction

Action :	1.0000
Comedy :	0.0143
Drama :	0.0000
Horror :	0.0000
Romance :	0.0000

(a) Bronson (2008).

Prediction

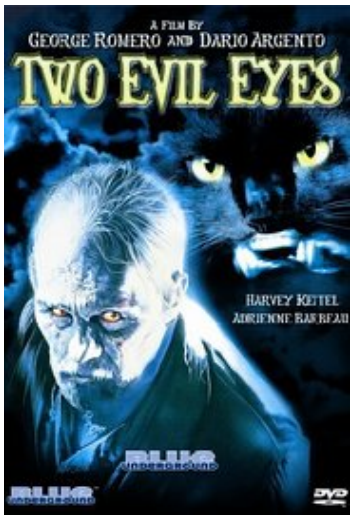
Action :	0.0000
Comedy :	1.0000
Drama :	0.0000
Horror :	0.0000
Romance :	1.0000

(b) Madhouse (1990).

Prediction

Action :	0.0000
Comedy :	0.0004
Drama :	1.0000
Horror :	0.0000
Romance :	1.0000

(c) Un Coeur en Hiver (1992).



Prediction

Action :	0.0000
Comedy :	0.0000
Drama :	0.0000
Horror :	1.0000
Romance :	0.0000

(d) Two evil eyes (1990).

Figure 10: Exemplary genre predictions.