deep net

Author and editors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Synonyms

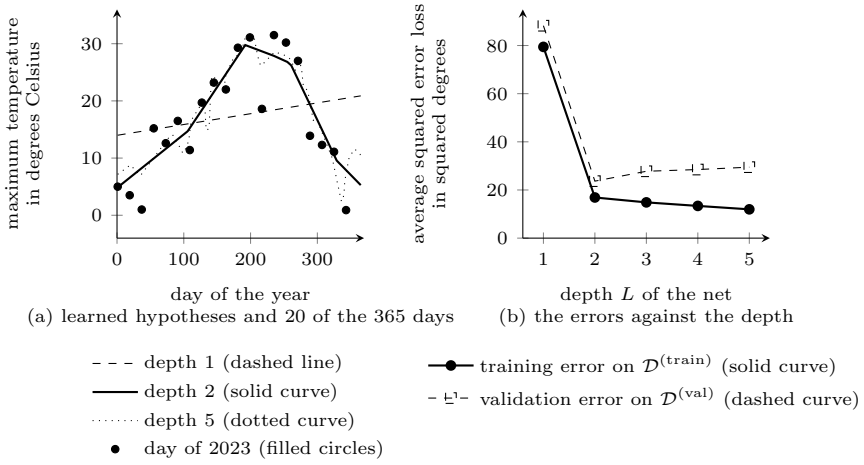
deep neural network

Abstract

A deep net is an artificial neural network (ANN) whose hypothesis maps are compositions of several layers, each of which applies an affine map and then an activation function to every entry of the result. The number of layers that carry learnable model parameters is the depth of the net, and deep learning is the name of the machine learning (ML) methods that use a deep net as their model. The layers before the last one compute a feature transformation learned from the data, and the last layer applies a linear hypothesis to the activations they deliver. A deep net therefore makes the choice of features part of training instead of a design decision taken beforehand. Its model parameters, the weights and offset terms of every layer, are learned by gradient descent (GD) on the training error, with the gradient computed by backpropagation. Depth is a property of the net, not of a single hypothesis: adding layers enlarges the hypothesis space and lowers the training error, while beyond some depth the validation error rises again, an instance of overfitting. Depth is therefore a hyperparameter, and a feature transformation chosen beforehand can reach a lower validation error with far fewer model parameters.

Definition

The daily maximum temperature at Krems an der Donau (Austria) rises from January to July and falls again towards December. The 365 days of 2023 form a training set $\mathcal{D}^{(\text{train})}$ and the 366 days of 2024 a validation set $\mathcal{D}^{(\text{val})}$, with the day of the year as the single feature of a day and its maximum temperature as the label. A hypothesis that is a linear function of the date can only rise or only fall over the year. Its training error, the average squared error loss over $\mathcal{D}^{(\text{train})}$, is 79.4 squared degrees, barely below the 83.4 of the constant hypothesis that always predicts the sample mean of the labels. A deep net replaces that single raw feature by features that it computes itself. A hidden layer of eight artificial neurons turns the date into eight activations, and a linear hypothesis of those eight numbers has a training error of 16.9 and a validation error of 23.7 on $\mathcal{D}^{(\text{val})}$ (Fig. 1(a)).



(a) learned hypotheses and 20 of the 365 days (b) the errors against the depth

Figure 1: Nets of depth 1 to 5 trained on the 365 days of 2023 at Krems, every hidden layer eight artificial neurons wide. (a) The learned hypotheses of depth 1 (dashed line), depth 2 (solid curve), and depth 5 (dotted curve) over the year, with 20 of the 365 days, every 18th day in calendar order (filled circles). (b) The training error (solid curve, filled circles) and the validation error on the 366 days of 2024 (dashed curve, open squares) against the depth. Data generated by `pythondemos/deepnet.py`

A deep net is an artificial neural network (ANN) whose hypothesis maps are compositions of L consecutive layers. Layer ℓ reads in the activations $\mathbf{a}^{(\ell-1)}$ of the previous layer, applies an affine map to them, and then an activation function σ to every entry of the result,

$$\mathbf{a}^{(\ell)} = \sigma\big(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)}\big) \text{ for } \ell = 1, \ldots, L-1, \tag{1}$$

starting from the raw feature vector $\mathbf{a}^{(0)} := \mathbf{x}$ of a data point. The last layer applies an affine map alone and returns the prediction

$$h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{b}^{(L)}. \tag{2}$$

The number L of layers that carry learnable model parameters is the depth of the net, and the term *deep learning* comes from it (Bishop and Bishop, 2024, Sect. 6.3; Goodfellow et al., 2016, p. 164). Deep learning covers the machine learning (ML) methods that use a deep net as their model. The activation function is prescribed rather than learned, most often the rectified linear unit (ReLU) $\sigma(z) = \max\{0, z\}$ (Goodfellow et al., 2016, Sect. 6.3). Without it, every layer would be an affine map, the layer recursion (1) and the output map (2) would compose to an affine map again, and the net would define the same hypothesis space as a linear model (Fig. 2).

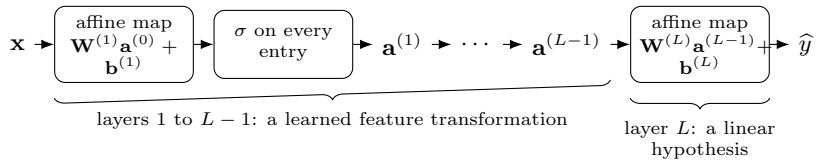


Figure 2: A deep net of depth L as a composition of layers. Each of the first $L-1$ layers applies an affine map to the activations of the previous one and then the activation function σ to every entry, and the last layer applies an affine map alone, so that the prediction $\widehat{y}$ is a linear hypothesis of the learned features $\mathbf{a}^{(L-1)}$

The composition splits at the last layer, and that split makes a deep net a feature learning method. The layers 1 to $L-1$ map the raw feature vector $\mathbf{x}$ to the activations $\mathbf{a}^{(L-1)}$, a feature transformation learned from the data, and the output map (2) applies an ordinary linear hypothesis to these learned features (Goodfellow et al., 2016, Sect. 6.1). A deep net therefore makes the choice of features part of training instead of a design decision taken beforehand. A deep net whose last hidden layer has two artificial neurons turns the date at Krems into two learned features. Least squares on those two reaches a training error of 14.5, against the 14.6 of the net's own last layer and the 79.4 of least squares on the raw feature. A learned feature transformation is often useful beyond the learning task it was trained for, so the layers before the last are kept and only the last layer is learned again on a smaller dataset (Bishop and Bishop, 2024, Sect. 6.3.4).

The model parameters $\mathbf{w}$ collect the weights in every matrix $\mathbf{W}^{(\ell)}$ and the offset terms in every vector $\mathbf{b}^{(\ell)}$, and a choice of $\mathbf{w}$ selects one hypothesis $h^{(\mathbf{w})}$ of the net. They are learned by empirical risk minimization (ERM): gradient descent (GD) with a fixed learning rate lowers the training error $\widehat{L}\big(h^{(\mathbf{w})}|\mathcal{D}^{(\text{train})}\big)$ by one gradient step after another. Each of them needs the gradient of the training error with respect to $\mathbf{w}$, and backpropagation computes it one layer at a time (Rumelhart et al., 1986).

Depth is a property of the net, not of a single hypothesis in it, and what a larger depth changes is the size of the hypothesis space. Fig. 1(b) shows the training error at Krems falling from 79.4 at depth 1 to 12.0 at depth 5, every hidden layer eight artificial neurons wide. The validation error does not follow it down. It is smallest at depth 2, at 23.7, and rises to 29.5 at depth 5, an instance of overfitting. Depth is therefore a hyperparameter, chosen on $\mathcal{D}^{(\text{val})}$ rather than fixed beforehand.

The depth-2 net lowers the validation error to 23.7 from the 88.2 of least squares on the raw feature. A feature transformation chosen beforehand reaches a lower validation error with a fifth of the model parameters. Least squares on the sine and cosine of the date at the annual period and at half of it needs five model parameters and has a validation error of 22.6, against the 23.7 of the net with its 25. Two further costs come with the net. Its training error

is not a convex function of $\mathbf{w}$ and has many local minima (Goodfellow et al., 2016, Sect. 8.2.2). GD therefore returns a different hypothesis from every initialization: five initializations of the depth-2 net give training errors between 16.9 and 19.1, and four of the five end above the smallest of them. The learned features also carry no reading of their own, where the sine and cosine name the annual cycle. An explainable artificial intelligence (XAI) method explains the trained net only by an approximation that cannot be fully faithful to it (Rudin, 2019, Sect. 2). A deep net is the answer where no such feature transformation can be written down beforehand: in image classification the first hidden layer delivers edges, the second corners and contours, and the third parts of objects (Goodfellow et al., 2016, Fig. 1.2).

Cross References

ANN, layer, activation, activation function, ReLU, backpropagation, deep learning, feature learning, feature transformation, overfitting, hyperparameter, interpretable ML, CNN, LLM

References

1. Bishop and Bishop (2024). Deep Learning: Foundations and Concepts. Springer Nature, Cham, Switzerland.

2. Goodfellow et al. (2016). Deep Learning. MIT Press, Cambridge, MA, USA.

3. Rumelhart et al. (1986). Learning Representations by Back-Propagating Errors. Nature, 323, pp. 533–536.

4. Rudin (2019). Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead. Nature Mach. Intell., 1(5), pp. 206–215.