What are linear models in deep learning?

The term linear model implies that **the model is specified as a linear combination of features**. Based on training data, the learning process computes one weight for each feature to form a model that can predict or estimate the target value.

Normalization is **a data pre-processing tool used to bring the numerical data to a common scale without distorting its shape**. Generally, when we input the data to a machine or deep learning algorithm we tend to change the values to a balanced scale.

Deep Learning models are creating state-of-the-art models on a number of complex tasks including speech recognition, computer vision, machine translation, among others. However, training deep learning models such as deep neural networks is a complex task as, during the training phase, inputs of each layer keep changing.

Normalization is an approach which is applied during the preparation of data in order to change the values of numeric columns in a dataset to use a common scale when the features in the data have different ranges. In this article, we will discuss the various normalization methods which can be used in deep learning models.

Let us take an example, suppose an input dataset contains data in one column with values ranging from 0 to 10 and the other column with values ranging from 100,000 to 10,00,000. In this case, the input data contains a big difference in the scale of the numbers which will eventually occur as errors while combining the values as features during modelling. These issues can be mitigated by [normalization](https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/normalize-data) by creating new values and maintaining the general or normal distribution in the data.

There are several approaches in normalisation which can be used in deep learning models. They are mentioned below

Batch Normalization

Batch normalization is one of the popular normalization methods used for training deep learning models. It enables faster and stable training of deep neural networks by stabilising the distributions of layer inputs during the training phase. This approach is mainly related to internal covariate shift (ICS) where internal covariate shift means the change in the distribution of layer inputs caused when the preceding layers are updated. In order to improve the training in a model, it is important to reduce the internal co-variant shift. The batch normalization works here to reduce the internal covariate shift by adding network layers which control the means and variances of the layer inputs.

Advantages

The advantages of batch normalization are mentioned below:

* Batch normalization reduces the internal covariate shift (ICS) and accelerates the training of a deep neural network
* This approach reduces the dependence of gradients on the scale of the parameters or of their initial values which result in higher learning rates without the risk of divergence
* Batch Normalisation makes it possible to use saturating nonlinearities by preventing the network from getting stuck in the saturated modes

Click [here](https://arxiv.org/pdf/1502.03167.pdf) to know more

Weight Normalization

Weight normalization is a process of reparameterization of the weight vectors in a deep neural network which works by decoupling the length of those weight vectors from their direction. In simple terms, we can define weight normalization as a method for improving the optimisability of the weights of a neural network model.

Advantages

The advantages of weight normalization are mentioned below

* Weight normalization improves the conditioning of the optimisation problem as well as speed up the convergence of stochastic gradient descent.
* It can be applied successfully to recurrent models such as LSTMs as well as in deep reinforcement learning or generative models

Click [here](https://papers.nips.cc/paper/6114-weight-normalization-a-simple-reparameterization-to-accelerate-training-of-deep-neural-networks.pdf) to know more

Layer Normalization

Layer normalization is a method to improve the training speed for various neural network models. Unlike batch normalization, this method directly estimates the normalisation statistics from the summed inputs to the neurons within a hidden layer. Layer normalization is basically designed to overcome the drawbacks of batch normalization such as dependent on mini batches, etc.

Advantages

The advantages of layer normalization are mentioned below:

* Layer normalization can be easily applied to recurrent neural networks by computing the normalization statistics separately at each time step
* This approach is effective at stabilising the hidden state dynamics in recurrent networks

Click [here](https://arxiv.org/abs/1607.06450) to know more

Group Normalization

Group normalization can be said as an alternative to batch normalization. This approach works by dividing the channels into groups and computes within each group the mean and variance for normalization i.e. normalising the features within each group. Unlike batch normalization, group normalization is independent of batch sizes, and also its accuracy is stable in a wide range of batch sizes.

Advantages

The advantages of group normalization are mentioned below:

* It has the ability to replace batch normalization in a number of deep learning tasks
* It can be easily implemented in modern libraries with just a few lines of codes

Click [here](https://arxiv.org/abs/1803.08494) to know more

Instance Normalization

Instance normalization, also known as contrast normalization is almost similar to layer normalization. Unlike batch normalization, instance normalization is applied to a whole batch of images instead for a single one.

Advantages

The advantages of instance normalization are mentioned below

* This normalization simplifies the learning process of a model.
* The instance normalization can be applied at test time.

**Dimensionality reduction** refers to techniques that reduce the number of input variables in a dataset.

More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.

High-dimensionality statistics and dimensionality reduction techniques are often used for data visualization. Nevertheless these techniques can be used in applied machine learning to simplify a classification or regression dataset in order to better fit a predictive model.

In this post, you will discover a gentle introduction to dimensionality reduction for machine learning

After reading this post, you will know:

* Large numbers of input features can cause poor performance for machine learning algorithms.
* Dimensionality reduction is a general field of study concerned with reducing the number of input features.
* Dimensionality reduction methods include feature selection, linear algebra methods, projection methods, and autoencoders.

**Kick-start your project** with my new book [Data Preparation for Machine Learning](https://machinelearningmastery.com/data-preparation-for-machine-learning/), including step-by-step tutorials and the Python source code files for all examples.

Let’s get started.

* **Updated May/2020**: Changed section headings to be more accurate.



A Gentle Introduction to Dimensionality Reduction for Machine Learning  
Photo by [Kevin Jarrett](https://flickr.com/photos/kjarrett/7647207998/), some rights reserved.

### Overview

This tutorial is divided into three parts; they are:

1. Problem With Many Input Variables
2. Dimensionality Reduction
3. Techniques for Dimensionality Reduction
   1. Feature Selection Methods
   2. Matrix Factorization
   3. Manifold Learning
   4. Autoencoder Methods
   5. Tips for Dimensionality Reduction

## Problem With Many Input Variables

The performance of machine learning algorithms can degrade with too many input variables.

If your data is represented using rows and columns, such as in a spreadsheet, then the input variables are the columns that are fed as input to a model to predict the target variable. Input variables are also called features.

We can consider the columns of data representing dimensions on an n-dimensional feature space and the rows of data as points in that space. This is a useful geometric interpretation of a dataset.

Having a large number of dimensions in the feature space can mean that the volume of that space is very large, and in turn, the points that we have in that space (rows of data) often represent a small and non-representative sample.

This can dramatically impact the performance of machine learning algorithms fit on data with many input features, generally referred to as the “[curse of dimensionality](https://en.wikipedia.org/wiki/Curse_of_dimensionality).”

Therefore, it is often desirable to reduce the number of input features.

This reduces the number of dimensions of the feature space, hence the name “dimensionality reduction.”

### Want to Get Started With Data Preparation?

Take my free 7-day email crash course now (with sample code).

Click to sign-up and also get a free PDF Ebook version of the course.

Download Your FREE Mini-Course

## Dimensionality Reduction

Dimensionality reduction refers to techniques for reducing the number of input variables in training data.

*When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the “essence” of the data. This is called dimensionality reduction.*

— Page 11, [Machine Learning: A Probabilistic Perspective](https://amzn.to/2ucStHi), 2012.

High-dimensionality might mean hundreds, thousands, or even millions of input variables.

Fewer input dimensions often mean correspondingly fewer parameters or a simpler structure in the machine learning model, referred to as degrees of freedom. A model with too many degrees of freedom is likely to overfit the training dataset and therefore may not perform well on new data.

It is desirable to have simple models that generalize well, and in turn, input data with few input variables. This is particularly true for linear models where the number of inputs and the degrees of freedom of the model are often closely related.

*The fundamental reason for the curse of dimensionality is that high-dimensional functions have the potential to be much more complicated than low-dimensional ones, and that those complications are harder to discern. The only way to beat the curse is to incorporate knowledge about the data that is correct.*

— Page 15, [Pattern Classification](https://amzn.to/2RlneT5), 2000.

Dimensionality reduction is a data preparation technique performed on data prior to modeling. It might be performed after data cleaning and data scaling and before training a predictive model.

*… dimensionality reduction yields a more compact, more easily interpretable representation of the target concept, focusing the user’s attention on the most relevant variables.*

— Page 289, [Data Mining: Practical Machine Learning Tools and Techniques](https://amzn.to/2tlRP9V), 4th edition, 2016.

As such, any dimensionality reduction performed on training data must also be performed on new data, such as a test dataset, validation dataset, and data when making a prediction with the final model.

## Techniques for Dimensionality Reduction

There are many techniques that can be used for dimensionality reduction.

In this section, we will review the main techniques.

### Feature Selection Methods

Perhaps the most common are so-called feature selection techniques that use scoring or statistical methods to select which features to keep and which features to delete.

*… perform feature selection, to remove “irrelevant” features that do not help much with the classification problem.*

— Page 86, [Machine Learning: A Probabilistic Perspective](https://amzn.to/2ucStHi), 2012.

Two main classes of feature selection techniques include wrapper methods and filter methods.

For more on feature selection in general, see the tutorial:

* [An Introduction to Feature Selection](https://machinelearningmastery.com/an-introduction-to-feature-selection/)

Wrapper methods, as the name suggests, wrap a machine learning model, fitting and evaluating the model with different subsets of input features and selecting the subset the results in the best model performance. RFE is an example of a wrapper feature selection method.

Filter methods use scoring methods, like correlation between the feature and the target variable, to select a subset of input features that are most predictive. Examples include Pearson’s correlation and Chi-Squared test.

For more on filter-based feature selection methods, see the tutorial:

* [How to Choose a Feature Selection Method for Machine Learning](https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/)

### Matrix Factorization

Techniques from linear algebra can be used for dimensionality reduction.

Specifically, matrix factorization methods can be used to reduce a dataset matrix into its constituent parts.

Examples include the eigendecomposition and singular value decomposition.

For more on matrix factorization, see the tutorial:

* [A Gentle Introduction to Matrix Factorization for Machine Learning](https://machinelearningmastery.com/introduction-to-matrix-decompositions-for-machine-learning/)

The parts can then be ranked and a subset of those parts can be selected that best captures the salient structure of the matrix that can be used to represent the dataset.

The most common method for ranking the components is principal components analysis, or PCA for short.

*The most common approach to dimensionality reduction is called principal components analysis or PCA.*

— Page 11, [Machine Learning: A Probabilistic Perspective](https://amzn.to/2ucStHi), 2012.

For more on PCA, see the tutorial:

* [How to Calculate Principal Component Analysis (PCA) From Scratch in Python](https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/)

### Manifold Learning

Techniques from high-dimensionality statistics can also be used for dimensionality reduction.

*In mathematics, a projection is a kind of function or mapping that transforms data in some way.*

— Page 304, [Data Mining: Practical Machine Learning Tools and Techniques](https://amzn.to/2tlRP9V), 4th edition, 2016.

These techniques are sometimes referred to as “manifold learning” and are used to create a low-dimensional projection of high-dimensional data, often for the purposes of data visualization.

The projection is designed to both create a low-dimensional representation of the dataset whilst best preserving the salient structure or relationships in the data.

Examples of manifold learning techniques include:

* [Kohonen Self-Organizing Map (SOM)](https://en.wikipedia.org/wiki/Self-organizing_map).
* [Sammons Mapping](https://en.wikipedia.org/wiki/Sammon_mapping)
* Multidimensional Scaling (MDS)
* t-distributed Stochastic Neighbor Embedding (t-SNE).

The features in the projection often have little relationship with the original columns, e.g. they do not have column names, which can be confusing to beginners.

### Autoencoder Methods

Deep learning neural networks can be constructed to perform dimensionality reduction.

A popular approach is called autoencoders. This involves framing a self-supervised learning problem where a model must reproduce the input correctly.

For more on self-supervised learning, see the tutorial:

* [14 Different Types of Learning in Machine Learning](https://machinelearningmastery.com/types-of-learning-in-machine-learning/)

A network model is used that seeks to compress the data flow to a bottleneck layer with far fewer dimensions than the original input data. The part of the model prior to and including the bottleneck is referred to as the encoder, and the part of the model that reads the bottleneck output and reconstructs the input is called the decoder.

*An auto-encoder is a kind of unsupervised neural network that is used for dimensionality reduction and feature discovery. More precisely, an auto-encoder is a feedforward neural network that is trained to predict the input itself.*

— Page 1000, [Machine Learning: A Probabilistic Perspective](https://amzn.to/2ucStHi), 2012.

After training, the decoder is discarded and the output from the bottleneck is used directly as the reduced dimensionality of the input. Inputs transformed by this encoder can then be fed into another model, not necessarily a neural network model.

*Deep autoencoders are an effective framework for nonlinear dimensionality reduction. Once such a network has been built, the top-most layer of the encoder, the code layer hc, can be input to a supervised classification procedure.*

— Page 448, [Data Mining: Practical Machine Learning Tools and Techniques](https://amzn.to/2tlRP9V), 4th edition, 2016.

The output of the encoder is a type of projection, and like other projection methods, there is no direct relationship to the bottleneck output back to the original input variables, making them challenging to interpret.

For an example of an autoencoder, see the tutorial:

* [A Gentle Introduction to LSTM Autoencoders](https://machinelearningmastery.com/lstm-autoencoders/)

### Tips for Dimensionality Reduction

There is no best technique for dimensionality reduction and no mapping of techniques to problems.

Instead, the best approach is to use systematic controlled experiments to discover what dimensionality reduction techniques, when paired with your model of choice, result in the best performance on your dataset.

Typically, linear algebra and manifold learning methods assume that all input features have the same scale or distribution. This suggests that it is good practice to either normalize or standardize data prior to using these methods if the input variables have differing scales or units.