MathWorks®

https://explore.mathworks.com/all-about-deep-learning-network-architectures

Deep Learning Questions Asked and Answered

By Johanna Pingel

All About Network Architectures: Choosing a Network Configuration

These articles will look at a topic, explain some of the background, and answer a few questions that I've heard from the MATLAB® community.

This column's topic is on deep learning network architectures. I've chosen questions that reflect common usage rather than academic use cases. I'll summarize the topic and then take a look at four questions:

1. What architecture should I use to classify images?
2. Can I take a pretrained CNN (trained for classification output) and train it for a regression problem?
3. How do I know which technique to choose for time-series regression?
4. Is there a network architecture I should use for a small dataset?

A network architecture defines the way in which a deep learning model is structured and more importantly what it's designed to do. The architecture will determine:

- The model's accuracy (a network architecture is one of many factors that impacts accuracy)
- What the model can predict
- What the model expects as input and output
- The combination of layers and how data flows through the layers

Most people start with someone else's combination of layers to begin training. When starting anything for the first time, it can be a good idea to look to the experts. Deep learning researchers have been exploring different network architectures and combinations of layers for quite some time. The results they have been getting with architectures like GoogLeNet, ResNet, or SqueezeNet (to name a few popular networks) are quite impressive. When first starting out, you can build on this success of a published architecture that tackles a similar problem rather than starting from scratch.

Deep Learning Toolbox™ supports many pretrained networks. See Pretrained Deep Neural Networks for a full list.

Before choosing a network architecture, it's important to understand what kind of use case you have and the common networks available to you.
You're likely to encounter these common architectures when getting started with deep learning:

- Convolutional neural network (CNN): CNNs are commonly associated with images as input data, but they can also be used for other input data, and I'll get into those details in question 1.

- Recurrent neural network (RNN): RNNs have connections that keep track of previous information to make future predictions. Unlike CNNs, where each input is assumed to be an independent event, RNNs can process sequences of data that might affect each other. One example is in natural language processing, where previous words influence the likelihood of what comes next.

- Long-short term memory (LSTM): LSTM networks are a commonly used RNN for sequence and signal data. I'll go into more detail in question 3.

- Generative adversarial network (GAN): Though I don't cover these in the questions below, GANs have become more popular most recently. GANs can generate new data based on existing data (think images of people that are not actually real people). I think it's exciting and a little futuristic; you can learn more about GANs in the How to Design and Train Generative Adversarial Networks (GANs) video and the Train Generative Adversarial Network (GAN) example.

Q1

I want a model to classify images. Which architecture should I use?

Excellent question. The short answer is you *probably* want a CNN to classify images.

Here's why.

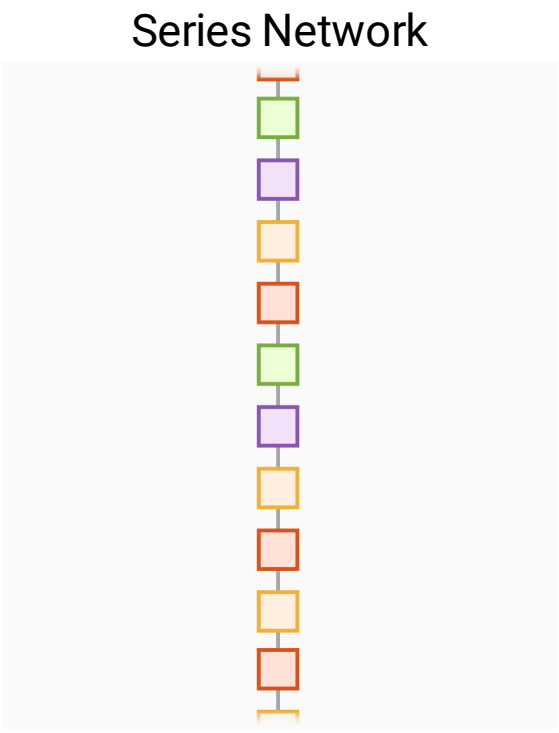
Let's start with what CNN and LSTM networks are, and how they are commonly used.

CNN

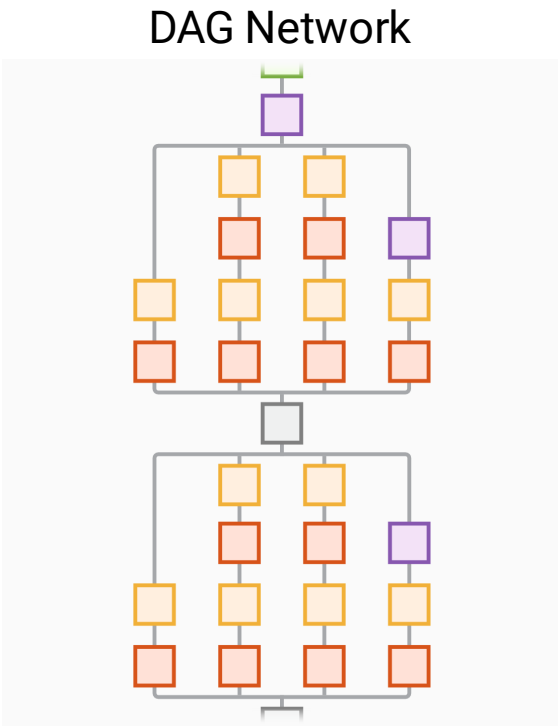
When talking about a convolutional neural network, some people say "ConvNet" but whenever I try dropping that in conversation, I always feel like I'm trying to be cool when I'm actually not.

CNNs consist of many layers but follow some semblance of a pattern of convolution | ReLU | pooling, which repeats again (and again and again). They are often useful for image classification because they are very good at local spatial pattern matching, and they also generally outperform other methods for image feature extraction. Keep in mind that at the heart of CNNs is **convolution**. Convolving the input image with a series of filters highlights the features in the image without losing spatial interaction between adjacent pixels.

There are many variations on this theme, but a few common configurations of CNNs are:



An example of AlexNet.
Series layers follow a straight line.



An example of GoogLeNet.
Multiple lines and connections
are a classic sign of DAG.

LSTM

Long short-term memory networks are primarily associated with time-series and sequence data. LSTM networks remember some portion of the data prior to making decisions now—they see data in context, which helps make better associations.

The rule of thumb is that time-series data is often best suited to LSTM networks and image data reliably works well with CNNs. Signal data is the exception that proves the rule. People use both CNNs and LSTM networks for signal data. I wrote an article on non-image applications of deep learning that includes an example where you could use a CNN for speech recognition.

This diagram illustrates a simple LSTM network for **classification**:

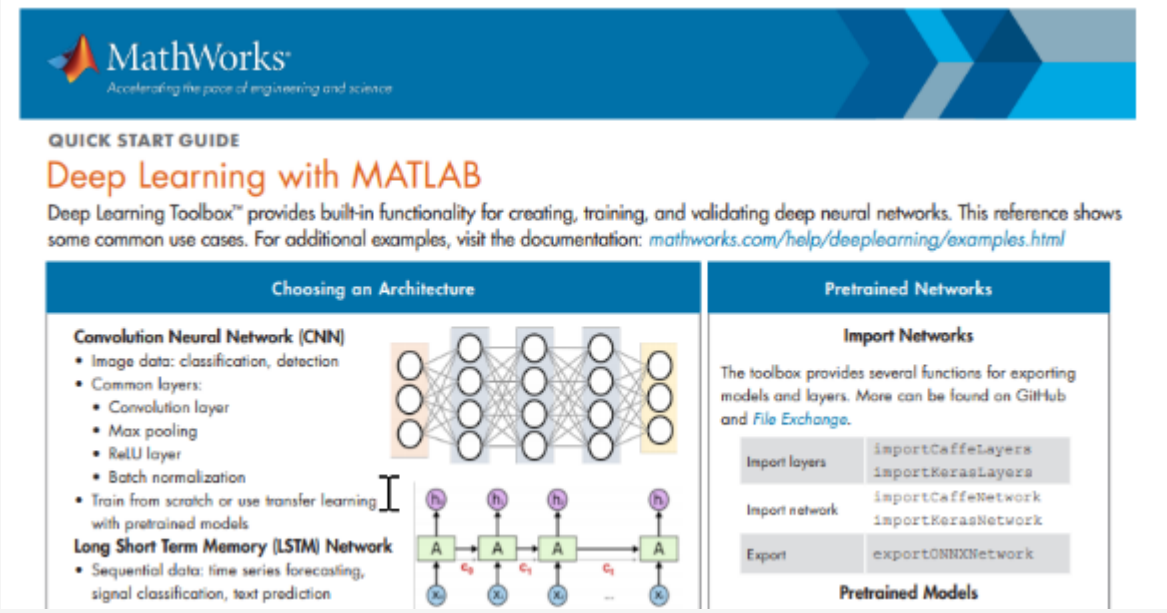
This diagram illustrates the architecture of a simple LSTM network for **regression**:

Q2

Can I take a pretrained CNN (trained for classification output) and train it for a regression problem?

Yes! You'll need to change the output layer from `classificationOutputLayer` to `regressionOutputLayer`, and there's a simple documentation example that will walk you through this process.

Here's a quick start guide for Deep Learning Toolbox that provides a handy reference for more of these kinds of questions.



Q3

There are so many options available for time-series regression! How do I know which architecture to choose?

My knee-jerk reaction is to always suggest LSTM networks! However, there are many techniques available for a reason, and some are going to work better in certain scenarios. I can't answer this question specifically without more information, so instead let's walk through a few possible scenarios.

Time-Series Regression Scenario #1*: My input is low-complexity time-series data. I have a series of data points that I want to use to forecast future events.*

In this situation, you might be best off using machine learning. Here's a short (3:43) video looking at forecasting electrical load using machine learning. Watch the video here.

Time-Series Regression Scenario #2: *I have data from multiple sensors and want to predict remaining useful life (the amount of time a machine has before it needs repair or replacement).*

My colleagues and I see this question with our customers in industrial automation who need to identify problems before they become dangerous or expensive. This time, you might want to use an LSTM network over machine learning regression. This approach reduces the need to identify features manually, which would be a significant task given multiple sensors.

Here's an example you can follow to see how to predict remaining useful life with an LSTM network.

Time-Series Regression Scenario #3*: I have audio data I want to denoise.*

Here you could use a CNN. The important thing about this method is to convert signals into images prior to passing them into the network. This means the signal becomes an image representation through Fourier transform or other time-frequency manipulation. Using images provides a way to see features you might not be able to visualize in the original

signal. The network used can be a pretrained network designed for images, since a Fourier transform is essentially an image.

Here's an example demonstrating how to denoise speech using a CNN.

Another thing I should mention for anyone in Scenario 3 is that ***wavelets*** are also a semipopular method to extract information from time-series data and then use that as input for a CNN. There's a nice write-up from researchers at UT Austin on how they converted brain signals to words and phrases using wavelets and deep learning.

Now, once again, you can do what you want here. It's very possible that you could also use an LSTM network in Scenario 1, or a CNN in Scenario 2. These scenarios are just meant to give you a starting point.

Q4

I want to build a classifier that will recognize images, but I have a limited dataset. Is there a network architecture that works better with a small dataset?

Network architecture and pretrained networks go hand-in-hand. A pretrained model is a neural network that has already undergone training. The weights and biases of the network are tuned to the input data, and the network can be retrained quicker for a new task. This process, called transfer learning, can sometimes require fewer images and work with a smaller dataset. An additional avenue to explore is to "create" more data through simulation or augmentation.

For more info on when to use a particular network architecture, we put together some tips and tricks that include information on pretrained networks as well.

For now, I will say that you should use whichever network you would like regardless of the dataset size, but consider using a pretrained network to require less input data, or consider methods to augment your dataset. My next column will cover pretrained networks and models, so keep an eye out for more on this topic.

Deep Learning Tips and Tricks

This page describes various training options and techniques for improving the accuracy of deep learning networks.

Choose Network Architecture

The appropriate network architecture depends on the task and the data available. Consider these suggestions when deciding which architecture to use and whether to use a pretrained network or to train fror

Data	Description of Task	Learn More
Images	Classification of natural images	Try different pretraine Deep Neural Networ To learn how to inter see Transfer Learnin
	Regression of natural images	Try different pretraine classification networ Regression Network.
	Classification and regression of non-natural images (for example, tiny images and spectrograms	For an example shov Classification. For an example shov Deep Learning.
	Semantic segmentation	Computer Vision Too segmentation. For m Learning (Computer
Sequences, time series, and signals	Sequence-to-label classification	For an example, see
	Sequence-to-sequence classification and regression	To learn more, see S Sequence Regressio