Chapter 5

An Overview of Deep Learning in Industry

Quan Le, Luis Miralles-Pechuán,
Shridhar Kulkarni, Jing Su, and Oisín Boydell

Contents

5.1 Introduction

Applications driven by deep learning are transforming our society. To name but a few examples: Google Assistant supports real-time speech-to-speech translation between 44 languages—helping to break down human communication barriers, Amazon Alexa understands human voice commands and assists in many everyday tasks such as ordering products and interacting with our smart homes, autonomous driving is assessed to be safer than human drivers, and deep learning models are helping us to understand our genome and to develop precision medicine.* Deep learning's ability to learn highly accurate representations of the task at hand, given enough annotated training data, helps it achieve better accuracy than traditional machine learning methods in many cases. This capability has opened up opportunities for its application to many new problems where traditional machine learning methods have fallen short.

In this chapter, we introduce the reader to many important applications of deep learning in industry. To begin, we provide a high-level overview of deep learning and its key architectures. We follow with a survey and discussion of the main applications of deep learning, categorized into four general tasks: recognition; generation; decision-making; and forecasting. We conclude the chapter with a discussion on the strengths and weakness, as well as the future applications of deep learning.

5.1.1 An Overview of Deep Learning

Deep learning models are artificial neural networks which emphasize the use of multiple connected layers (modules) to gradually transform input signals to the desired outputs. Given a sufficiently large data set of input-output pairs, a training algorithm can be used to automatically learn the mapping from the inputs to the outputs by tuning a set of parameters at each layer in the network. The input data is typically left in its raw form—for example, the gray level values for the pixels in an image or the raw readings over time from a set of sensors. Once the optimal values for the parameters at each layer in a network have been learned, we can view the layers as encoding high-level features extracted from the raw input data.

* www.forbes.com/sites/insights-intelai/2019/02/11/how-machine-learning-is-crafting-precision-medicine/

As such, deep learning models do two things at the same time: learning an effective feature representation from the raw inputs and learning a model that maps from the feature representation to the outputs. The early layers of the models may only capture simple features calculated from the raw inputs, but the later layers combine these simple features to learn more abstract features which are optimized for the task at hand.

5.1.1.1 Deep Learning Architectures

Many different deep learning architectures or configurations of connected layers and arrangements of connections between layers have been proposed. In this section, we present the main deep learning architectures from the literature (Goodfellow et al. 2016). They are the main components of the deep learning models used in the applications we discuss later on.

- *Convolutional Neural Network*. Convolutional neural networks layers (CNNs) (LeCun et al. 1989) are neural network layers designed to process data whose features have regular spatial dependency (e.g., the grid-like topology of images, or other multi-dimensional data). CNNs take advantage of these dependencies by applying local *convolutional* filters in specialist layers in the network. A CNN module is typically composed of a succession of convolutional layers. A CNN network typically connects inputs to a CNN module, then to a fully connected feed-forward module that ultimately produces the outputs of the network (Figure 5.1).

  The early convolutional layers in a CNN typically learn low-level local features, such as edges and lines in the case of image data, while later layers with bigger receptive fields will combine these local features into more complicated features, such as shapes or even faces (such as in facial recognition). As the same stationary pattern could appear anywhere in the raw input, the same set of filters should be applied everywhere in the input. This feature of CNNs is called *parameter sharing*, helping it avoid the *overfitting* problem. The characteristics of CNNs have made them incredibly effective at image processing tasks, as well as other tasks involving low-level inputs with spatial dependencies with which they can take advantage.
- *Recurrent Neural Network Modules*. Recurrent neural network modules (RNN modules) (Rumelhart et al. 1986) are a family of neural network architectures designed to process sequential data. RNN modules allow looped connections through which the hidden state calculated from the last input presented to an RNN module is included as the input to the RNN module along with the next set of input values so as to encode sequential relationships in a network.

  Theoretically, RNNs can handle long-term dependencies and use information of arbitrary long sequences, but in practice this is not the case due to the gradient vanishing and gradient explosion problems. Long short term

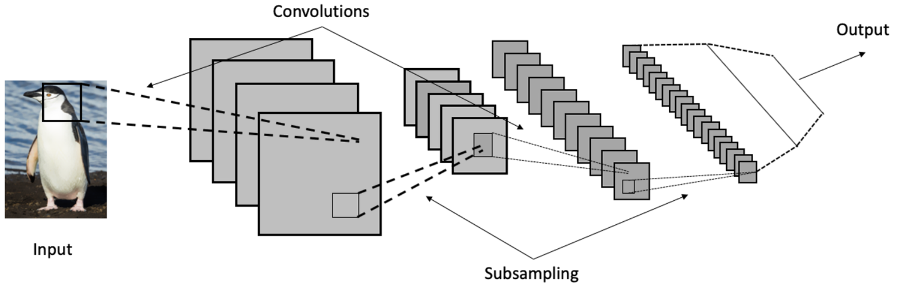

Figure 5.1 A typical convolutional neural network model.

memory (LSTM) networks (Hochreiter and Schmidhuber, 1977) and other gated architectures have been proposed as a way to alleviate the problems with earlier RNN architectures, and have been hugely effective in many applications from speech recognition, to machine translation, to finding machine faults based on sensor data (see Schmidhuber (2015) for a detailed overview of the applications of LSTMs).

- *Residual Architecture*. Depth in a deep learning model is essential for its expressivity and its generalization performance. For example, from 2012 to 2015, the top-5 error (a classification of an image is considered as an error if the correct label is not in the top 5 predicted categories) in the 1000-class image classification problem of the ImageNet challenge reduced from 16.4% to 3.57% as the depth of the neural networks increased from 8 to 152 layers. While convolutional neural layers are effective in exploiting the correlations between data features to create new useful features, they still face the same problem as other multiple layer networks: having gradients that either explode or diminish after being propagated through multiple layers. The residual network module is designed to address this problem (Kaiming et al. 2015). Residual architectures have since been used extensively for a wide range of applications.
- *Attention Module*. The attention module arose in the context of the sequence to sequence models (Sutskever et al. 2014) used for machine translation—where an RNN encoder network is used to convert the original text sequence into a context vector, and an RNN decoder network is used to translate the context vector to the piece of text in the target language, one word at a time. Researchers realized that the original sequence to sequence model performs badly for long texts, hence Bahdanau et al. (2014) proposed using the attention module on the hidden state sequence of the encoder to provide context for the decoder at each generation step. Nowadays, deep models using attention (e.g., Transformer (Baltrušaitis et al. 2018), BERT (Devlin et al. 2018)) is the dominant approach in natural language processing, as well as in other sequence learning problems.

5.1.2 Deep Generative Models

Generating new content is an important area of machine learning with applications ranging from conversational artificial intelligence (AI) to knowledge discovery, and generative models are approaches that simulate how the data from the desired distribution are generated. Once a generative model is learned from a data set of samples, it could be used for generating new data as well as for other important inferencing tasks. Deep generative models (Goodfellow et al. 2016) refer to the neural network based generative models; in this section, we will discuss three major classes of deep generative models with many important applications: autoregressive models, a variational autoencoder, and a generative adversarial network.

Deep autoregressive models use neural networks to generate future data given past data in a chosen direction; they have no latent random variables.

Autoencoder is the unsupervised neural network approach to learn how to encode data efficiently in a latent space. An autoencoder includes two neural networks: an encoder that maps data from the input space to the latent space and a decoder to reconstruct the input data from the encoded latent data. A variational autoencoder (VAE) is a class of autoencoder where the encoder module is used to learn the parameter (mean, standard deviation) of a distribution. And the decoder is used to generate examples from samples drawn from the learned distribution.

A generative adversarial network (GAN) includes two components: a generator network and a discriminator network; a data set of examples from the desired source (e.g., the images of dogs) is required to train the GAN. The generator generates a simulated example given as its input a latent variable value drawn from a specified distribution. Both the generated examples and the authentic ones are fed to the discriminator network, whose job is to distinguish between the authentic and the simulated examples. The GAN is trained by updating the weights of the discriminator by gradient descent to increase its discriminative power, while updating the weights of the generator by gradient ascent to improve its ability to mimic the authentic examples and fool the discriminator. Over time the generator will learn to generate new data which simulate well the examples drawn from the target source.

5.1.3 Deep Reinforcement Learning

Reinforcement learning (RL) is a machine learning branch aimed at solving problems in which a set of sequential decisions is needed to maximize a goal. RL has a completely different approach to supervised and unsupervised learning. The goal of supervised learning is to create models that learn patterns from a set of labeled data to generate a function that maps the entries with the output. Whereas the purpose of unsupervised learning is finding hidden structures within unlabeled data. On the other hand, the RL algorithm goal is to learn a set of sequential actions that maximize the cumulative reward based on the experience obtained when interacting with the environment (e.g., playing a game, manipulating a robot, or activating and deactivating the heater). Some examples of RL objectives are playing online games at a human level, driving cars without human intervention, or managing traffic lights to reduce traffic.

RL was developed a few decades ago but, because of some limitations, it was unable to reach its full potential. This situation changed in recent years with the development of deep learning. Deep learning algorithms, for example, the popular deep Q-learning algorithm is able to approximate the Q-Table with an artificial neural network very effectively in terms of memory and processing requirements. The combination of deep learning and RL is called deep reinforcement learning (DRL) and has multiplied the potential of RL, boosting the general interest in RL of the industry and the scientific community.

5.2 Applications of Deep Learning

As the effectiveness of deep learning approaches has become evident, they have become widely applied. In this section, we provide an overview of some notable applications of deep learning. We categorize applications of deep learning as achieving one of four distinct objectives: *recognition*, *generation*, *decision-making*, and *forecasting*. Throughout this section, we will see how applications of deep learning across many different industries and domains rely on a small set of the same core deep learning approaches (Figure 5.2).

5.2.1 Recognition

In the context of machine learning, and specifically deep learning, *recognition* is defined as a task of identifying predefined labels from unannotated inputs—for

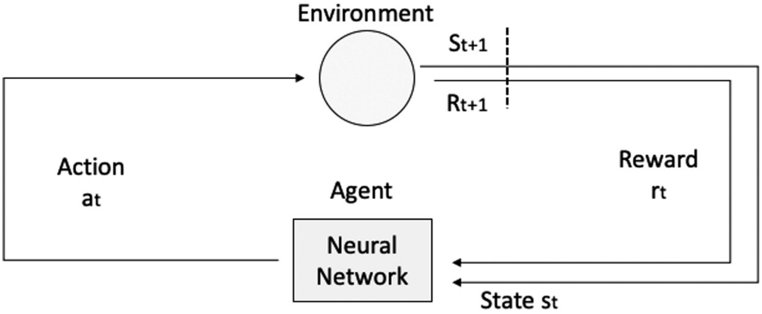

Figure 5.2 In reinforcement learning, there is an agent that takes actions in an environment. The environment gives a reward for that action and the agent moves to the next state.

example, recognizing that an image contains a particular species of flower—and is an example of a *supervised learning* task. Deep learning received much of its early attention because of its success in recognizing objects in images and so we describe this application in detail. Recognition, however, is not confined to image-based applications. Deep learning approaches have improved the accuracy of object recognition in videos, audio, and text as well as other types of data and we describe key applications of those.

5.2.1.1 Recognition in Text

Much of the digital data we collect and analyze is text-based (e.g., scientific literature, news articles, contracts, and medical reports), and the popularity of social media has led to an online text explosion. Deep learning has been applied successfully in natural language processing (Collobert et al. 2011) where deep feed-forward neural networks (often with the use of convolution layers) are used to learn features automatically and perform competitively in key tasks along the natural language processing pipeline.

Deep text models are the dominant approach for sentiment analysis, usually a part of any text analytics pipeline. Modern sentiment analysis systems can be decomposed into different subtasks: target detection and polarity classification (Zhang et al. 2018). Li et al. (2017) proposed deep memory networks for both of these tasks. Other text information extraction tasks that have benefited from the application of deep learning include topic classification, semantic role labeling, part of speech tagging, named entity recognition, and chunking (Young et al. 2018).

In the era of *fake news*, important work is being done on using deep neural networks—usually a combination of CNN and RNN models—to detect deceptive opinions (Sharma et al. 2019). In a recent development, Graph CNN was used to detect fake news on social media with high accuracy (Monti et al. 2019). FakeNewsAI* is an example of a commercial news verification service based on a deep learning architecture.

Another commercial application of a deep text model is in the Google search engine, where the attention based BERT model is used to rank results returned from a search query.† In an example of a more end-to-end application of deep learning, Amazon's Alexa personal assistant uses LSTM to classify text and recognize the commands of the users (Naik et al. 2018). Apple's Siri‡ and Google's Assistant§ also use similar techniques.

* www.fakenewsai.com

† https://www.blog.google/products/search/search-language-understanding-bert/

‡ www.apple.com/ios/siri/

§ assistant.google.com

5.2.1.2 Recognition in Audio

Making sense of audio has long been a focus of the machine learning community, and deep learning has made significant progress in this field which leads to important commercial applications. Conversational speech recognition (CSR) is one of the most important such applications. In 2016, a series of breakthroughs were reported in CSR based on the advances of deep neural networks. For example, IBM researchers (Saon et al. 2016) achieved a 6.6% error rate with a large vocabulary English conversational telephone CSR system using a combination of CNNs and RNNs. In 2016, Microsoft achieved the milestone of reaching human parity in speech recognition for English with a 5.9% error rate on a similar task (Xiong et al. 2016). For commercial use cases, Google uses speech recognition to generate subtitles for videos automatically as well as to carry out voice searches.

Speech recognition is a more challenging task in noisy environments or in conversations. Different from single-speaker speech recognition, a step of vocal source separation is needed. For example, in order to process recordings from a cocktail party, we need to identify the voice from one speaker out of other speakers as well as background noise. Simpson et al. (2015) have applied CNNs to this problem to very good effect.

Moving away from speech, identifying the characteristics of music has been widely studied in machine learning and again deep learning is making inroads here. For music recommendation systems, automatic music genre recognition is a key step and deep learning brings a new approach in this area. Julien Despois demonstrated how to classify the genre of a piece of music or a complete song with CNNs* achieving genre recognition accuracies in the high 90% range. Niland is an important commercial player in the application of deep learning to music and in 2017 was acquired by Spotify.†

5.2.1.3 Recognition in Video and Images

Deep learning approaches, and in particular CNNs, are especially well suited for processing visual data. The large image data set ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015), where the 2012 version includes 1,432,167 images labeled in 1,000 classes through crowd sourcing, has been the driving force for the development of new deep learning architectures in computer vision. Deep learning announced its arrival in computer vision by winning the ILSVRC image classification task in 2012 with the AlexNet CNN model: AlexNet achieved a top-5 error rate of 15.3%, significantly better than the error rate of the best non-deep learning models of 26.5%. After it, CNN models have become the dominant approach for computer vision tasks, and in 2015 the Residual

* chatbotslife.com/finding-the-genre-of-a-song-with-deep-learning-da8f59a61194

† www.spotify.com

Network (He et al. 2015) achieved the landmark of performing better than human performance. Figure 5.3 shows an example of image classification on a CNN model trained on the ImageNet data set.

Similar CNN-based approaches have been used for image recognition tasks such as Google photo search for medical application (Hegde et al. 2019), image caption generation, and video frame classification.[*] Salesforce research arm[†] is an example of a commercial application that uses deep models for textual sentiment analysis, as well as image classification tasks. There are also good examples of deep learning-based image recognition solutions being used to drive revenue in niche areas. For example, Deepomatic[‡] have leveraged deep learning to build commercially successful image tagging services in domains including fashion, security, and interior design.[§] Similarly, Tractable[¶] are using deep learning to estimate the cost of repair for insurance claims by recognizing the amount of damage in images of cars involved in incidents.

Face recognition is a long-standing image processing challenge. Prior to the introduction of deep learning models, state-of-the-art approaches to face recognition in images relied on first recognizing a set of carefully selected hand-crafted features within an image using image processing techniques, and then using these as an input to a machine learning model. CNNs enable very accurate facial recognition in an end-to-end system. For example, the DeepFace system from Facebook (Taigman et al. 2014) is a nine-layer deep CNN-like neural network used for face recognition. The DeepFace system was shown to achieve an accuracy of 97.35% on the Labeled Faces in the Wild (LFW) data set (Huang Erik Learned-Miller, 2014) (a well-known and challenging face recognition benchmark), which is significantly better than the state-of-the-art non deep learning approaches prior to 2014. Subsequent to the release of DeepFace, Google researchers introduced FaceNet (Schroff et al. 2015), which was also based on CNNs and achieved 99.63%

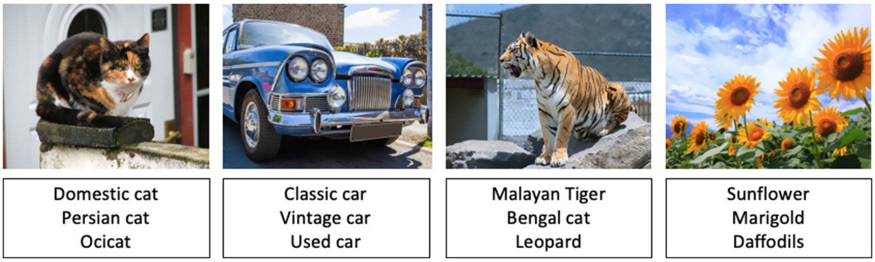

Figure 5.3 Examples of image classification

[*] cs.stanford.edu/people/karpathy/deepvideo/

[†] https://einstein.ai/

[‡] www.deepomatic.com

[§] www.deepomatic.com/demos

[¶] www.tractable.ai/technology

accuracy on the LFW data set. Face recognition is an active area of research with many new deep learning models being proposed recently; for example, SphereFace and ArcFace (Wang and Deng, 2018). It has also been widely applied in business: from mobile systems that implement the technique to log users in automatically, to many law enforcement agencies who use it to detect criminals.

The success of deep learning models in image classification has translated to many successful applications of deep models in medical images, often through the use of the transfer learning technique. One notable example is the system described by Ciresan et al. (2013) that won the MICCAI 2013 Grand Challenge on Mitosis Detection. The mitosis detection task is particularly interesting as unlike the ILSVRC image classification task (Russakovsky et al. 2015), the goal is not to classify an entire image as belonging to a category, but rather to identify the portions of a large image that belong to a particular category—in this case, examples of mitosis in an image. This is referred to as image segmentation and is at the core of many image processing tasks. In recent years, deep models have attained human expert level performance in multiple tasks, including melanoma screening and detection, identifying diabetic retinopathy, cardiovascular risk analysis, and pneumonia detection on chest X-rays (Esteva et al. 2019).

The vast amounts of aerial imagery enabled by the lowering costs in satellite technology and the prevalence of low-cost aerial drones underpin another area in which deep learning-based image recognition is being widely applied. In 2010, Mnih and Hinton (Mnih and Hinton, 2015) produced a pioneering work in which deep learning methods were used to identify roads in aerial images. This is an example of an image segmentation problem (similar to the medical image recognition problem described previously) as the system not only recognizes that a road is present in an image but also the specific pixels in the image that contain the road. Marcu (Marcu, 2016) used CNNs to very accurately segment portions of aerial images into semantically meaningful categories (e.g., roads, buildings, parks). Recently, Facebook used deep learning to automatically segment roads from satellite images and generate accurate maps.* Their work automates the costly and time-consuming process of annotating maps manually, which will be helpful in many unmapped regions of the world—especially the regions in developing countries.

Aiming for the insurance industry, TensorFlight† uses these techniques to analyze aerial images and to provide automatic annotation on construction type, building footprint, and roof characteristics. Terrapattern‡is an interesting example of a group adopting similar deep learning-based approaches to build an aerial photo search engine that will find common patterns in massive collections of aerial

* ai.facebook.com/blog/mapping-roads-through-deep-learning-and-weakly-supervised-training/

† www.tensorflight.com

‡ www.terrapattern.com/about

imagery. Figure 5.4[*] shows an example set of search results for wastewater treatment plants.

Extending systems that recognize objects in images to systems that recognize objects in video is an obvious step and modifications to the core deep learning approaches (e.g., CNNs) to work on video have been shown to work well (Girshick, 2015; Ren et al. 2015). Clarifai[†] is an interesting startup working on automatic object recognition in video using deep learning for a wide range of tasks in industry. They are especially focused on the advertising industry and they use their technology to find appropriate videos in which to place ads.

Deep learning has allowed a step change in the performance of systems built to recognize objects in images and are now the *de facto* standard for that task. CNNs can be used both to classify entire images or to segment objects within images. It is worth noting that object recognition in *non-standard images* remains very challenging. For example, while it is possible (e.g., Valdenegro-Toro, 2016), object recognition in the sonar images collected by autonomous underwater vehicles (AUVs), widely used in the oil and gas industry, remains very difficult for automated systems. Similarly, it is worth noting that almost real-time object recognition is required in certain applications (e.g., autonomous vehicle control). While this can be achieved in some cases (e.g., Iandola et al. [2016] for traffic light recognition), it remains a significant challenge in using deep learning as significant computation is required to use a deep network to make a prediction. GANs are also beginning to become

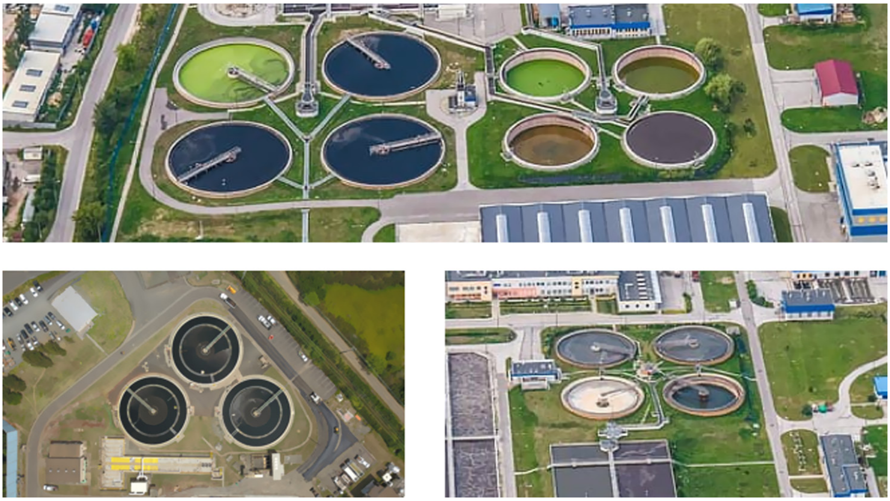

Figure 5.4 GeoVisual search results for wastewater treatment plants in satellite imagery.

[*] www.medium.com/descartestech/geovisual-search-using-computer-vision-to-explore-the-earth-275d970c60cf

[†] www.clarifai.com

widely adopted for image recognition tasks but only for specialist applications and CNNs remain much more popular.

5.2.2 Content Generation

Rather than recognizing what is available in the data, in *generation* tasks, the objective is to output novel or additive content based on input data. Examples include generating captions for images, converting a piece of music into a new style (Hadjeres and Pachet, 2016), or composing entire documents. This section surveys key applications of deep learning for generating novel content. The ability to build machine learning systems that generate new content is something that did not really exist before the advent of deep learning approaches and has spurred a renewed interest in the area of *computational creativity*.

5.2.2.1 Text Generation

In terms of using deep learning for generation, text has attracted more attention than any other data format. Deep learning approaches have been successfully applied to many different tasks including generating captions for images, generating conversational responses for chatbots, generating screenplays, novels or speeches, and machine translation. In this section, we describe a selection of interesting examples of this application of deep learning technology.

Image and video captioning techniques are created to address the multimodality challenge of visual data and human language. A good captioning model needs to identify key objects in view and output a fluent sentence showing correct relations between the identified objects. Basically, there are two approaches in the research community including end-to-end pipelines and stepwise captioning models.

The *Show and Tell* caption generator from Google researchers (Vinyals et al. 2016) gives an early example of end-to-end captioning pipelines. A CNN network is employed to *encode* an input image to a fixed-length vector. Thereafter, this vector is taken as the initial hidden state of an RNN network. RNN *decodes* the vector into a sentence. There is no object detection step in this pipeline; loss of caption error is counted from each generation step. The *Show and Tell* model features ease of manipulation and quality of reading experience.

It was the winner of the Microsoft COCO 2015 Image Captioning Challenge.*

The *NeuralTalk* model sets a milestone of stepwise captioning practice (Karpathy and Li, 2015). This model is not a single CNN plus an RNN. Instead, a more complicated module is applied to extract visual features. The authors use a region convolutional neural network (R-CNN) (Girshick et al. 2014) to detect object regions from an input image. This R-CNN was pre-trained on ImageNet. Thereafter, an image-sentence score metric is introduced to find the maximum correspondence

* mscoco.org/dataset/#captions-leaderboard

between each object region and a word in caption sequence. In 2016, Karpathy et al. released NeuralTalk2, a revision of the original system capable of more believable captions.[*] While the current state-of-the-art of these captioning systems is not yet capable of human-level performance, these systems are already being applied in commercial offerings; for example, automatic captioning of images in the Facebook newsfeed.[†]

Going further than simple image caption generation, the movie director Oscar Sharp and the AI researcher Ross Goodwin developed *Benjamin*,[‡] an LSTM-based system that can generate original screenplays automatically. This is achieved by training it with dozens of science fiction screenplays and then asking it to generate its own. Their system was capable of generating long sections of novel movie script—one of which was actually filmed and released as the short film *Sunspring*.[§]

Although most of the text generation systems described so far are commercially interesting, they have not yet seen wide industrial adoption. *Machine translation* of texts from one language to another, on the other hand, is of massive commercial value. Deep learning approaches to the machine translation task, commonly referred to as neural machine translation (NMT), have led to a step change in the performance of automated machine translation systems. Instead of using phrase-level matching between two languages (as is done in older approaches to machine translation), the NMT model works on entire sentences which provide NMT systems with the opportunity to model more contextual information than is possible in other approaches. Google's NMT system is a good example of a modern NMT engine and it has three main components: the *encoder LSTMs*, the *decoder LSTMs*, and an *attention module* (Wu et al. 2016). The encoder LSTMs transforms an input sentence to a list of vector representations with one vector per symbol. The decoder LSTMs takes the vectors from the encoders and generates one language symbol at a time. The attention module regulates the decoders to focus on specific regions during decoding to drive increased accuracy of translations, and their addition was an important step in driving translation accuracy.

NMT systems reach translation error rates significantly below those statistical machine translation (SMT) approaches. As a result, Facebook have moved their entire translation system to an NMT-based solution based on LSTMs which will handle more than 2,000 translation directions and six billion translations per day. Skype by Microsoft has also deployed an NMT-based translation system. In this case, speech is automatically translated from one language to another. The system first performs speech recognition on the original language, then translates the text to the destination language, before finally using a text-to-speech system to generate speech in the destination language, where all of these components rely on deep

[*] cs.stanford.edu/people/karpathy/neuraltalk2/demo.html

[†] www.wired.com/2016/04/facebook-using-ai-write-photo-captions-blind-users/

[‡] bigcloud.io/filming-the-future-how-ai-directed-a-sci-fi-short/

[§] www.arstechnica.com/the-multiverse/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/

learning models. Skype translator currently supports speech-to-speech translation between ten languages.*

5.2.2.2 Audio Generation

It is also possible to use deep learning approaches to generate audio. Speech synthesis is by far the most studied application but approaches to music composition and sound effect generation have also been proposed. In this section, we describe some of the most interesting applications of deep learning approaches to audio generation.

Generating authentic sounding artificial speech, or *speech synthesis*, has long been a focus of artificial intelligence researchers. Deep neural networks, however, bring new approaches to this long-standing challenge. WaveNet (van den Oord et al. 2016), a deep autoregressive model developed by Google DeepMind, achieves state-of-the-art performance on text-to-speech generation and the generated speech audio is rated as subjectively natural by human raters. This performance is achieved with a dilated CNN model that manages to model long-term temporal dependencies with a much lower computational load than LSTM models.

Recently, text-to-speech synthesis techniques reached a new milestone after the landmark of WaveNet (van den Oord et al. 2016), and Google researchers introduced Tacotron 2 (Shen et al. 2017). This system employs a sequence-to-sequence model to project textual character embeddings to spectrograms in the frequency domain. Then a modified WaveNet model generates time-domain waveform samples from spectrogram features. Compared with WaveNet, Tacotron 2 has a better performance in learning human pronunciations and its model size is significantly smaller.

Beyond text-to-speech (TTS) techniques, speech-to-speech (STS) has drawn attention in recent years. Google researchers introduced a direct STS translation tool, named as Translatotron (Jia et al. 2019). Traditionally, speech-to-speech translation is achieved in three steps (or models) including speech-to-text transcription on the source language, text-to-text translation, and text-to-speech synthesis to generate audio in the target language. This routine is well established with convincing accuracy, also it is widely deployed in commercial applications. Translatotron is the first trial to merge the aforementioned three steps in one model and show its value. Although the benchmark of Translatotron is slightly below a baseline model on the Spanish-to-English translation task, this direct translation approach is able to mimic the voice of the source speaker in the synthesized target speech.

As a side-effect of the advances on TTS, it is now easy to generate a fake voice or speech toward a target person. An AI startup Dessa released a speech synthesis model called RealTalk which creates the human voice perfectly.† Currently, details of data set, models, and benchmarks are not publicly available, but people can try to tell the real voice from the fake on this page.‡

* www.skype.com/en/features/skype-translator/

† medium.com/dessa-news/real-talk-speech-synthesis-5dd0897eef7f

‡ http://fakejoerogan.com/

Rather than generating speech from text, deep learning approaches have also been used to generate sound effects based on video inputs. An artificial foley artist* described by Owens et al. (2015) can reproduce sound effects for simple silent videos based on an ensemble of CNN and LSTM models. A CNN model is trained to extract high-level image features from each video frame. A sequence of these image features (color and motion) is taken as input to an LSTM model, and the LSTM model is trained to create an intermediate sound representation known as a *cochleagram*. In the final step, the *cochleagram* is converted to waveforms through an LSTM-based sound synthesis procedure. Although only applied in very simple environments, the results are impressive.

Deep learning models can also be used to generate original music. DeepBach (Hadjeres and Pachet, 2016), for example, uses an LSTM-based approach to *compose* original chorales in the style of Bach. The model is composed of multiple LSTM and CNN models that are combined in an ensemble which given a melody can produce harmonies for the alto, tenor, and bass voices. Similar systems based on RNNs that generate original music in other styles have also been demonstrated—for example, music in the style of Mozart[†,‡] or traditional Irish music.[§,¶]

5.2.2.3 Image and Video Generation

Deepfake is a buzz word in the recent news press. This word comes from *deep learning* and *fake*. Paul Barrett, adjunct professor of law at New York University, defines deepfake as falsified videos made by means of deep learning. We would like to confine the concept of deepfake as falsified human faces in image or video made by generative adversary networks (GAN) (Goodfellow et al. 2014) or related AI techniques. The general goal of deepfake is to transfer stylistic facial information from reference images or videos to synthetic copies.

Hyperconnect** released MarioNETte, one of the state-of-the-art face reenactment tools in 2019 (Ha et al. 2019). Previous research suffers from identity preservation problems on unseen large poses. MarioNETte integrates image attention block, target feature alignment, and landmark transformer. These modifications lead to better realistic synthetic videos.

Other than research publications, we find face reenactment tools for smartphones. ZAO, a free deepfake face-swapping app, is able to place user's face

[*] vis.csail.mit.edu

[†] www.wise.io/tech/asking-rnn-and-ltsm-what-would-mozart-write

[‡] www.hochart.fr/rnn/

[§] highnoongmt.wordpress.com/2015/08/07/the-infinite-irish-trad-session/

[¶] highnoongmt.wordpress.com/2015/05/22/lisls-stis-recurrent-neural-networks-for-folk-music-generation/

[**] https://hyperconnect.com/?lang=en

seamlessly and naturally into scenes from hundreds of movies and TV shows using just a single photograph.*

Deepfake techniques are developing fast and this is becoming a challenge for personal privacy and public security. It is not only humans that cannot tell a faked portrait or video clip from the original, but advanced face recognition software is also being cheated. Korshunov and Marcel (Korshunov and Marcel, 2018) performed a study where the results showed that state-of-the-art recognition systems based on VGG and Facenet neural networks are vulnerable to Deepfake videos, with 85.62% and 95.00% false acceptance rates respectively. The best fake detection method is based on visual quality metrics which shows an 8.97% error rate on high-quality Deepfakes.

In order to improve fake detection techniques, Korshunov and Marcel (2018) released the first public available fake video data set, vidTIMIT. Tech giants also joined this campaign. Recently, Google and collaborators released a deep fake detection data set with over 3,000 manipulated videos (Rössler et al. 2019). Facebook and partner organizations started the Deepfake Detection Challenge (DFDC) and funded over US$10 million to support this industry-wide effort.†

Image generation refers to the process of automatically creating new images based on existing information sources. Deep learning has been applied in many image generation tasks, including image (and video) super-resolution, image colorization, image generation from text or other images, and so-called neural art.

Image super-resolution (ISR) is an image generation problem in which the resolution of a digital image is vastly increased through the application of algorithms. In recent years, Microsoft researchers have applied CNNs to this problem and achieved state-of-the-art restoration quality (Dong et al. 2014). Although deep CNNs significantly improve the accuracy and speed of ISR, there still remains a challenge of restoring the finer texture details. Ledig et al. (Ledig et al. 2016) proposed SRGAN for image super-resolution. The SRGAN is capable of restoring photo-realistic natural images for 4× upscaling factors. Recently, a team from ElementAI developed HighRes-net,‡ a deep learning model capable of stitching multiple low-resolution satellite images to create a super-resolution image. Unlike other super-resolution models which could add fake details to the final image, their model recovers the original details in the super-resolution version after aggregating the information from multiple low-resolution ones. As such, their model has wide applications: from automatic land management to mapping road networks.

While it remains a very challenging task (and performing it at a human level is well beyond the current state-of-the-art), deep learning has led to advances in the ability of systems to automatically generate images based on textual descriptions. Systems that can do this can be helpful in graphic design, animation, and

* www.theverge.com/2019/9/2/20844338/zao-deepfake-app-movie-tv-show-face-replace-privacy-policy-concerns

† ai.facebook.com/blog/deepfake-detection-challenge/

‡ www.elementai.com/news/2019/computer-enhance-please

architecture. RNNs are one of the successful approaches to automatically synthesizing images from texts. Mansimov et al. (2015) introduced a seminal approach to image generation. There are two parts in the model by Mansimov et al. A bidirectional RNN is used to learn the sequence (or alignment) of words in input captions. Another generative RNN is used to learn the sequence of image patches from training images. Mansimov's model successfully generates synthesized images from input captions and some of the images are novel from the training set. However, the generated images often look blurry and need further refinement.

More recently, GANs have been demonstrated to be useful for image generation from text. Reed et al. (2016) introduced a text-conditional convolutional GAN architecture to address this challenge. In this design, both the generator network and the discriminator network use convolution layers for text encodings. The GAN generated images tend to look more natural than those produced using other methods.

Slightly different from generating images from text, it is also possible to generate new images from existing ones. For example, there are massive numbers of pictures captured by Google's Street View project, but an image from a required point of view may not be available. To solve this problem, Google researchers proposed DeepStereo (Flynn et al. 2015), in which CNN models are trained to predict new views based on available image sources to a quite good effect.* Similarly, the Irish company Artomatix† uses models based on CNNs to generate realistic looking texture for 3D models based on existing images.

Framing image generation as an image-to-image translation problem, Isola et al. (2016) used conditional adversarial networks to generate photo-realistic images from edge maps or sketches.‡ Zhu et al. (2016) proposed generative visual manipulation methods for similar objectives to create more stylized images.§ Going even further away from photo-realistic images, so called neural art seeks to create stylistic representations of images. For example, Gatys et al. (2015) used CNNs to generate new paintings with template artistic styles. A sample output image is shown in Figure 5.5.

5.2.3 Decision-Making

The recognition and generation systems described in previous sections perform niche tasks that are often embedded in larger systems. It is, however, also possible to build end-to-end control systems using deep learning, in particular, DRL which is, as described in Section 5.1.3, the combination of deep learning and reinforcement learning.

* For examples of DeepStereo see www.youtube.com/watch?v=cizgVZ8rjKA

† www.artomatix.com

‡ Christopher Hesse has a demonstration of Isola's model at www.affinelayer.com/pixsrv/

§ A demonstration is available at people.eecs.berkeley.edu/~junyanz/projects/gvm/

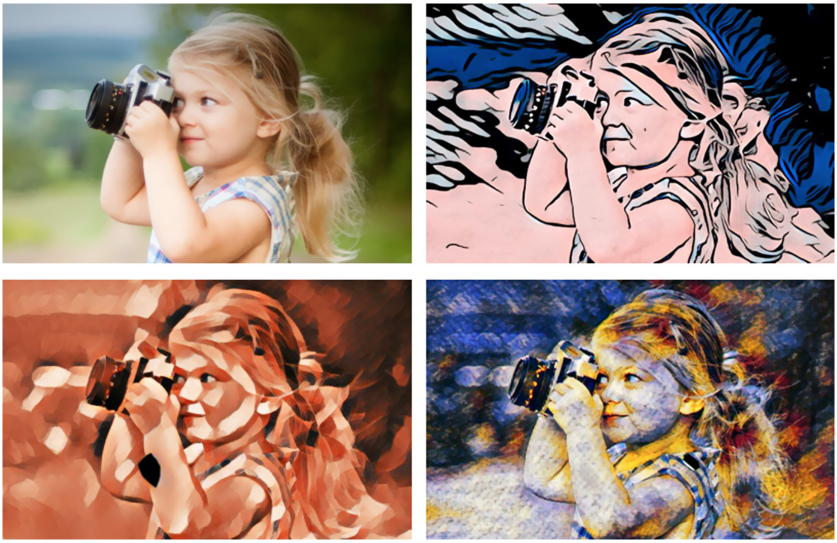

Figure 5.5 **An example of neural art. (Reproduced from www.instapainting.com.)**

In this subsection, we introduced systems mainly based in DRL that make decisions continuously in dynamic and static environments. As it has been described by Yuxi Li (2017), there is a wide range of applications areas where DRL can be effectively applied such as dialogue systems, education, healthcare, or computer vision. In the following sections, we focus on applications in autonomous driving, game playing, robotics, energy consumption, online advertising, and finance.

5.2.3.1 Autonomous Driving

An autonomous car or self-driving car is "a vehicle that is capable of sensing its environment and navigating without any human input" (Hussain, 2016). Deep learning approaches are often used for object recognition as part of an autonomous driving pipeline, and technologies based on these systems dominate the current commercial autonomous driving efforts. Google's self-driving car unit, for example, started in 2009 and in the next seven years drove over two million miles of test journeys on open roads. This car implements deep learning models extensively for object recognition. Similarly, the Tesla Autopilot system incorporates Tesla-developed deep learning systems based on CNNs for the tasks of recognizing objects through vision, sonar, and radar sensors. There are also examples of smaller startup self-driving car companies such as Drive.ai, which created a deep learning-based software for autonomous vehicles, or Tealdrones.com, a startup that equips drones with onboard deep learning modules for image recognition and navigation.

In this section, however, we are more interested in DRL based end-to-end control systems in which deep learning models are used not only for recognition tasks but also to actually make control decisions based on inputs from cameras and other sensors. The use of neural networks for end-to-end autonomous driving control has a history that stretches back to the late 1980s. The ALVINN system (Pomerleau and Pomerleau, 1989) or Nvidia's DAVE-2 system are good examples of a modern deep learning approach to controlling autonomous cars. In the case of the DAVE-2 system, CNN models generate steering commands based on the video input from three cameras placed at the front of the car. To train the model, long segments of recorded video, together with human steering, were used to link the correct steering commands with the camera inputs. The DAVE-2 system has achieved impressive results in simulation and has driven long journeys in a fully autonomous mode. Deep Tesla is another interesting example of an end-to-end autonomous car control system based on CNNs. However, due to some limitations, it is much more likely that, for the moment, deep learning models will be used for developing specific components of self-driving car control systems such as pedestrian detection or road-sign recognition.

As described in Section 5.1.3, RL creates a table that associates states to actions. In such a way that the agent driving the car is constantly looking up the table to see which is the best action for each state. RL cannot be applied in training mode in real scenarios because the agent will take random actions to learn and this can be very dangerous (Sallab et al. 2017). This is a big handicap for real-world applications because it is not possible in real life to have accidents in order to learn. However, there are simulations of the environment in which the car can learn how to behave and once the car learns the right actions, it can be deployed in the real world. For example, in an investigation carried out by Dong Li et al. (2018), neural networks were trained with the images obtained from a car simulator called TORCS (The Open Racing Car Simulator). The implemented model used to drive the car was composed of two modules: one based on multitasking neural networks, which were responsible for taking the driver's vision as an input, and another one based on DRL, which was responsible for making decisions from the extracted features of the neural network. This proposed model was capable of driving the car with great precision and was also able to adapt to new screens (situations) not previously seen by the system. The lack of real-world simulators is one of the main limitations of implementing DRL end-to-end systems in cars, nevertheless, the number of cars driving with multiple cameras makes it much more feasible to create a real-world simulator where DRL can be trained.

5.2.3.2 Automatic Game Playing

In 2015, Google DeepMind received massive publicity for its DRL system that could play Atari 2600 games at a superhuman level (Van Hasselt et al. 2016). This performance was achieved through the use of deep Q-networks (DQNs), which is an approach that combines both deep neural networks and reinforcement learning (Van Hasselt et al. 2016). DQNs incorporate a CNN model trained to predict the action with the highest expected reward based on an image of the current game state. The reward is the mechanism by which RL algorithms learn. For example, in Atari 2600 games, the agent gets a positive reward when it increases the score and a negative reward when the agent loses a game. In the case of the DQNs applied to the Atari 2600 (Van Hasselt et al. 2016), it was notable that it only utilized an image of the game screen as the input. The agent was able to learn by itself and it achieved superhuman levels after playing the game for several hours.

Deep Q-Networks can be distinguished from other deep learning architectures by the fact that they do not require labeled training sets. Rather, training DQNs involves multiple iterations of experimentation, the success of which is measured using an appropriate reward function. For the DeepMind Atari game playing system, the reward function was the score achieved in a round of playing the game. By playing hundreds of thousands of rounds, the system used the reward to modify the parameters of a deep network and to guide it to a version that could achieve superhuman performance at the game.

The DQN algorithm inspired many researchers to develop control systems for other games. For example, MIT researchers developed DeepTraffic, a gamification of highway traffic, Lin applied a DQN to play FlappyBird,* and Tampuu et al. (2015) applied a DQN to a multiagent Pong game where each agent was controlled by a DQN.

It is also worth including board games among the applications discussed here for two reasons: first, the level of difficulty that they entail, and second if a DQN is able to solve difficult problems, it will also be able to solve the easy ones. The problem of making a machine able to play chess or the game of Go better than a human has been a challenge for AI since its beginning (Silver et al. 2018). Some remarkable authors such as Alan Turing and John Von Neumann tried to develop hardware and software to enable computers to play board games. One of the milestones of AI was achieved when the Deep Blue program beat the world chess champion in 1997. However, the development of these programs required great levels of supervision by experts in both chess and coding (Figure 5.6).

The most widely known example of a deep learning approach to play board games is probably that of DeepMind's AlphaGo, which is an autonomous agent for the game of Go. AlphaGo defeated the world Go champion, Lee Sedol, 4-1 in March 2016 (Chouard, 2016) and continues to beat world-class human players. The AlphaGo model uses deep convolutional neural networks and a general tree search algorithm (Silver et al. 2017). The architecture of AlphaGo is especially interesting as it is a hybrid system incorporating Monte Carlo tree search algorithms, supervised learning (SL) policy networks, reinforcement learning policy networks, and value networks (Silver et al. 2016). The first of these components, Monte Carlo tree search, has been a mainstay of automated Go playing systems since the 1990s (Brugmann, 1993). The latter three components are implemented

* https://github.com/yenchenlin/DeepLearningFlappyBird

Figure 5.6 Algorithms based on deep reinforcement learning techniques have surpassed human level performance in many games such as chess (left), video games (center), and Go (right).

as CNN models with slightly different objectives. This makes AlphaGo an interesting mix of traditional and cutting edge AI techniques.

AlphaZero, which was developed following on from the work on AlphaGo, leverages RL so that it does not need any human telling the agent what the best movements are or which are the best strategies. Through RL, the agent is able to learn by itself on the basis of trial and error. Playing against itself over many millions of games, the agent is able to learn which are the best moves in many different games, including chess and the game of Go.

5.2.3.3 Robotics

While autonomous vehicles can be thought of as robots, there is a much broader set of robotic systems that also implement deep learning approaches to control their actions. We are primarily interested in systems that use deep learning for end-to-end robot control rather than systems that use deep learning components for specific tasks such as object recognition or speed control. For example, BRETT* from UC Berkeley can learn to perform tasks such as stacking LEGO blocks, putting together a toy plane, and screwing bottle caps onto bottles using deep reinforcement learning (Levine et al. 2015). CNN models are used to process the image input and to convert them into motor control signals. The model is trained using a technique similar to the DQN approach previously described (Levine et al. 2015). Other similar systems include the Google Brain grasping robot (Levine et al. 2016) and the systems developed by Delft Robotics that used deep learning to win the Amazon Picking Challenge.†

Moving away from fixed-base grasping robots, Peng et al. (2016) introduced a deep reinforcement learning approach for a robot locomotion control policy in a physical simulation. The objective in this simulation setting was to navigate a simulated robot dog through different types of terrain obstacles, such as gaps, walls, and slopes. This control task shares similar challenges to those faced in the Atari

* https://news.berkeley.edu/2015/05/21/deep-learning-robot-masters-skills-via-trial-and-error/

† http://amazonpickingchallenge.org/

2600 games previously described. The terrain descriptions, as well as the robot state descriptions, are high dimensional, therefore they are not suitable to be used directly for traditional reinforcement learning systems (Figure 5.7).

In this case, a mixture of actor-critic experts (MACE) approach is introduced to learn the right output, the parameterized actions, for complex input data (e.g., leaps or steps) (Peng et al. 2016). The actor-critic approach in RL consists of creating two models: the first takes the actions and the latter evaluates how good the action was, so the agent can learn. The MACE system is a type of CNN in which the inputs are terrain features and the robot state features, and the outputs are the actions and estimated rewards.

As we already said for self-driving cars, it is unlikely that in the short term, end-to-end deep learning systems will be used for the control of autonomous robots that interact closely with humans. Rather, it is more likely that deep learning models will be used as components in these systems for specific tasks such as object recognition or text detection. For robotic applications that do not involve significant interactions with people, however, end-to-end control systems based on deep reinforcement learning are feasible. Another interesting potential for the application to robotics is using simulations to train models that are deployed in real robots (Rusu et al. 2016). This overcomes the complication of performing hundreds of thousands of experiment iterations in a real environment.

5.2.3.4 Energy Consumption

One of the applications that called the attention of many companies has been the reduction by 40% of the cooling bill of Google data centers.* These data centers, which can be seen in Figure 5.9, are in charge of storing all the information collected in Google applications such as emails in Gmail, photos in Google Maps, or

Figure 5.7 Deep reinforcement learning techniques have been successfully used in robotic arms (left) and in drones (right).

* https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40

the documents in Google Drive. This information is stored in large rooms with around 2.5 million servers, according to the company Gartner Inc. To prevent these servers from overheating, it is necessary to cool them down every so often using a cooling system usually based on pumps, chillers, and cooling towers.

Developing an algorithm to optimize energy consumption to cool down the servers is very complex because each data center has its own characteristics. These include factors such as the local climate, the demand for computing power, and the cost of energy. However, any improvement in the reduction of energy consumption can result in a large economic saving and a potential reduction in energy generation derived carbon emissions.

Google decided to address the problem of maximizing energy consumption by implementing an algorithm based on deep reinforcement learning. The objective of the algorithm is to maximize the power usage effectiveness (PUE), which is a metric obtained by dividing the total energy consumed by the data center by the energy consumed to run the computer infrastructure. Google does not usually give information about the techniques used to achieve their achievements, and the case of maximizing the PUE was not an exception. However, Google itself published information showing that its algorithm was trained with historical data from thousands of sensors that captured information from several factors (temperature, consumption, pump speed, etc.) to create a system able to automatically adapt to dynamic environments by using deep neural networks (Figure 5.8).

5.2.3.5 Online Advertising

Bidding optimally in internet advertising is a very complex task where RL has been successfully applied (Du et al. 2017). In the paper presented by Han Cai et al. (2017), a case is presented of success where an agent is able to bid intelligently for each impression in the real-time bidding (RTB) model. RTB is like a massive worldwide auction where publishers offer visits (from users who access their pages) to advertisers so that they can display their ads. Generally, in online

Figure 5.8 **An approach based on reinforcement learning has reduced the electricity bill of Google Data Centers by 40%.**

advertising, bids are made by algorithms on behalf of the advertiser because it would be impossible to individually bid on each impression (there could be thousands per second).

Generally, the objective of RTB algorithms is to get the largest number of clicks from a certain budget. However, in other approaches, different metrics are used such as the number of conversions or the revenue generated (Zeff and Aronson, 1999).

Usually, if the bidding price is very high, the budget will be finished quicker, and the number of impressions will be lower. On the other hand, if the bidding price is too low, the number of impressions will be very small because other candidates with a higher price will be selected. Additionally, we have to consider that the market is in constant movement. Advertisers raise and lower the price all the time. And advertisers and publishers come and go. It is, therefore, a very complex problem. To test the performance of RL to address this problem, RL algorithms were compared with linear bidding strategy (LIN) (Perlich et al. 2012), which is the best state-of-the-art algorithm for this problem, with the RL algorithm giving a higher performance.

5.2.4 Forecasting

In this section, we introduce the application of deep learning to forecasting future values of a time series. We distinguish between three main categories of application—forecasting physical signals, forecasting financial data, and forecasting wind speed and power.

5.2.4.1 Forecasting Physical Signals

Physical signals in the real world are complex and inter-correlated. Although the variation of one environmental condition may not affect people's lives if it is not on an extreme level, there is a requirement in many industries to precisely predict the values of one or more physical signals into the near future. For example, forecasts of *solar irradiance* (the power per unit area received from the sun) have long been of interest to the industry. Generally, solar irradiance on a day t+1 can be modeled with three types of inputs: the value of solar irradiance on day t or before and the values of other physical factors (e.g., air temperature, humidity, wind speed, wind direction, sunshine duration, or geographical location). Mellit and Pavan (2010) applied a wavelet network deep learning model to predict solar irradiance 24 hours into the future. This model achieves forecasting accuracies in the high 90% range and is at the top end of the current state-of-the-art. Kmet and Kmetova (2015) introduced an application of a specific type of RNN known as an echo state network (ESN) (Jaeger, 2001) for the same problem with similarly impressive results. Cao and Lin (Cao and Lin, 2008) proposed a diagonal recurrent wavelet neural network to forecast global solar irradiance and proved that the model is capable of mapping solar irradiance, which is usually highly non-linear and time changeable, as it combines advantages of recurrent and wavelet neural networks (Figure 5.9).

Another application is in wind forecasting in relation to wind energy production. Wind prediction is complex due to the wind's high degree of volatility and deviation. Therefore, in real electricity markets, system operators have barely begun to factor wind forecast information into their daily operations and reserve determination. However, in terms of academic research, many publications have introduced short-term or long-term wind forecasting technologies and experience based on deep learning approaches. In Kariniotakis et al. (1996), a recurrent higher-order neural network (RHONN) model was developed for wind power forecasting in a wind park. This model can be used to predict wind speed or power in time scales from some seconds to three hours. The work of More and Deo (2003) employs the technique of neural networks (feed-forward and recurrent networks) and statistical time series respectively to forecast daily and monthly wind speeds in India. The results show that the neural networks perform better than the baseline ARIMA models. However, the average of daily and monthly wind speed could be smoother than that of hourly wind speed, which implies that it is not difficult to obtain a more accurate forecasting result for daily and monthly wind speed. In Rohrig and Lange (2006), a method based on artificial neural networks was utilized to predict the average hourly wind speed. A multilayer perceptron neural network with three-layer feed-forward architecture was adopted as their forecasting system. The input selection was determined on the basis of correlation coefficients between previous wind speed observations.

Other authors of deep learning applications to physical system forecasting include: Romeu et al. (2013) who applied *stacked denoising autoencoders* (Vincent et al. 2010) (a specific type of feed-forward network) to the task of indoor temperature forecasting and James et al. (2017) who applied deep learning models to forecast wave conditions. These techniques are also being applied in industry. For

Figure 5.9 Deep learning has been widely applied in both domains, wind speed estimation (left) and forecasting financial data (right).

example, Nervana* leveraged deep learning for extreme weather forecasting and climate change modeling and Descartes Labs† applied deep learning over satellite images for predicting crop production.

5.2.4.2 Forecasting Financial Data

Reviewing the application of deep learning to financial forecasting is difficult as commercial players are very secretive about the techniques that they use and how effective they are. There are, however, a smaller number of examples of academic publications illustrating how this can be done. Currency exchange rate forecasting is one that has received attention, and one that is recognized as a challenging problem (Beirne et al. 2007). Chao et al. (2011) used *deep belief networks*, a form of feed-forward network, to accurately predict future exchange rates. Galeshchuk and Mukherjee (2017a, 2017b) report good results applying recurrent neural networks to the same problem.

Not surprisingly, researchers have also focused on attempting to predict stock prices, with some success.

Bao et al. (2017) used a combination of autoencoder networks and recurrent neural networks to predict stock prices with impressive results. Similarly, Fischer and Krauss (2017) demonstrated how recurrent networks can be used effectively for market predictions. Rather than focusing on predicting prices directly, a lot of work focuses on volatility–accurate predictions which are key for devising trading strategies. It has been shown repeatedly that hybrid models including recurrent networks and feed-forward networks can be used to do this effectively (Poklepović et al. 2014; Kristjanpoller et al. 2014; Monfared and Enke, 2015; Lu et al. 2016).

One of the advantages of deep learning approaches is their ability to handle sparse data types. News sources are essential driving forces for stock market activities and may be even more influential than the current and past stock prices (Fama, 1965). Ding et al. (2015) proposed a CNN-based framework to model the influence of news events on stock market prices. The proposed framework has two key features. First, a neural tensor network (NTN) is applied to learn event embeddings from word embeddings in an automatic way. Second, a CNN model covering both short-term and long-term events is used to predict a binary output indicating whether a stock price is rising or falling. This CNN-based architecture is demonstrated to show increased prediction accuracy over state-of-the-art non-deep learning baseline methods (Ding et al. 2015). A later modification to this approach (Ding et al. 2016) integrates knowledge graph information during the learning process of event embeddings to predict stock prices even more accurately.

* www.nervanasys.com/

† www.descarteslabs.com/

As mentioned previously, detailed reports of how deep learning is being used for commercial applications are difficult to find. However, many new companies are open about the fact that they are trying to perform such actions. Sentient Technologies* led by Babak Hodjat formerly of Apple, for example, is a hedge fund that puts deep learning approaches at the heart of their trading strategies.† Similarly, Man Group,‡ one of the world's largest hedge funds, is utilizing deep learning methods extensively in their trading strategies.§ A recent report by EurekaHedge¶ analyzed a range of machine learning-based hedge funds and showed that, in general, they are outperforming other types of funds.

Numerai** is a particularly interesting player in the financial forecasting space. Rather than creating their own forecasting models, Numerai has created a platform through which interested data scientists can access data sets and deploy their own forecasting models. The forecasts made by this disparate set of models are then combined into an ensemble on which a hedge fund is based. With a payment system built on top of blockchain technologies, Numerai is an interesting experiment in crowdsourcing investment decisions from deep learning experts that could be worth watching.††

5.3 Conclusion

In this chapter, we have reviewed a broad range of different applications of deep learning, categorized into four general types of task: recognition, generation, decision-making, and forecasting, with a particular focus on relevance to industry. As we have shown, deep learning has produced impressive achievements both in terms of improvements in accuracy, compared with traditional machine learning approaches, as well as enabling completely new AI applications. We are already benefiting from many of these in our everyday lives, and as applications of deep learning continue to improve and expand, we can expect to experience many more benefits in the future.

However, recent applications of deep learning in industry have also raised concerns such as hidden biases in training data, adversarial manipulation of trained models, and the difficulty in understanding the rationale behind decisions made by

* www.sentient.ai

† www.bloomberg.com/news/articles/2017-02-06/silicon-valley-hedge-fund-takes-on-wall-street-with-ai-trader

‡ www.man.com/

§ www.bloomberg.com/news/features/2017-09-27/the-massive-hedge-fund-betting-on-ai

¶ www.eurekahedge.com/Research/News/1639/Quantitative-Hedge-Funds-Strategy-Profile

** www.numer.ai

†† www.wired.com/2017/02/ai-hedge-fund-created-new-currency-make-wall-street-work-like-open-source/

deep learning models due to their black box nature. Further research is needed to make deep learning applications safer and more trustworthy in society.

References

D. Bahdanau, K. Cho and Y. Bengio. "Neural machine translation by jointly learning to align and translate". *arXiv preprint arXiv:1409.0473* (2014).

T. Baltrušaitis, C. Ahuja and L.-P. Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE*, 2018.

W. Bao, J. Yue and Y. Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory". *PloS one* 12.7 (2017), e0180944.

J. Beirne, J. Hunter and M. Simpson. "Is the real exchange rate stationary? – A similar sized test approach for the univariate and panel cases". In: *Economics and Finance, Dept of Economics and Finance Research Papers, Brunel University* (2007). url: http://bura.brunel.ac.uk/ handle/2438/1024.

B. Brugmann. *Monte Carlo go*. Tech. rep. Technical report, Physics Department, Syracuse University, Syracuse, NY, 1993.

H. Cai et al. "Real-time bidding by reinforcement learning in display advertising". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 661–670.

J. Cao and X. Lin. "Application of the diagonal recurrent wavelet neural network to solar irradiation forecast assisted with fuzzy technique". *Engineering Applications of Artificial Intelligence* 21.8 (2008), pp. 1255–1263.

J. Chao, F. Shen and J. Zhao. "Forecasting exchange rate with deep belief networks". In: *The 2011 International Joint Conference on Neural Networks*. July 2011, pp. 1259–1266. doi: 10.1109/IJCNN. 2011.6033368.

T. Chouard. "The Go Files: AI computer wraps up 4-1 victory against human champion". *Nature News* (2016).

Dan C. Cireşan et al. "Mitosis detection in breast cancer histology images with deep neural networks". In: *D. Image Computing and Computer-Assisted Intervention – MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22–26, 2013, Proceedings, Part II*, Springer, Berlin Heidelberg, 2013, pp. 411–418. isbn: 978-3-642-40763-5. doi: 10.1007/978-3642-40763-5 51.

R. Collobert et al. "Natural language processing (almost) from scratch". *Journal of Machine Learning Research* 12(Aug 2011), pp. 2493–2537.

J. Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805* (2018).

X. Ding et al. "Deep learning for event-driven stock prediction". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. AAAI Press, Buenos Aires, Argentina, 2015, pp. 2327–2333. isbn: 978-1-57735-738-4. url: http://dl.acm.org/citation.cfm?id=2832415.2832572.

X. Ding et al. "Knowledge-driven event embedding for stock prediction". In: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. 2016, pp. 2133–2142. url: http://aclweb.org/anthology/ C/C16/C16-1201.pdf.

C. Dong et al. "Learning a deep convolutional network for image super-resolution". In: *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part IV.* Ed. by David Fleet et al. Springer International Publishing, Cham, 2014, pp. 184–199. isbn: 978-3-319-10593-2. doi: 10.1007/978-3-319-10593-2 13.

M. Du et al. "Improving real-time bidding using a constrained markov decision process". In: *International Conference on Advanced Data Mining and Applications*. Springer, 2017, pp. 711–726.

A. Esteva et al. "A guide to deep learning in healthcare". *Nature Medicine* 25.1 (2019), pp. 24–29.

E. F. Fama. "The behavior of stock-market prices". *The Journal of Business* 38.1 (1965), pp. 34–105. issn: 00219398, 15375374. url: http://www.jstor.org/stable/2350752.

T. Fischer and C. Krauss. *Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions*. Tech. rep. FAU Discussion Papers in Economics, 2017.

J. Flynn et al. "DeepStereo: Learning to predict new views from the world's imagery". *ArXiv* abs/1506.06825 (2015). url: http://arxiv.org/abs/1506.06825.

S. Galeshchuk and S. Mukherjee. "Deep learning for predictions in emerging currency markets." *ICAART* 2. 2017a, pp. 681–686.

S. Galeshchuk and S. Mukherjee. "Deep networks for predicting direction of change in foreign exchange rates". *Intelligent Systems in Accounting, Finance and Management* 24.4, (2017b), pp. 100–110.

L. A. Gatys, A. S. Ecker and M. Bethge. "A neural algorithm of artistic style". *ArXiv e-prints* (Aug. 2015). arXiv: 1508.06576 cs.CV.

R. B. Girshick. "Fast R-CNN". *arXiv* abs/1504.08083 (2015). url: http://arxiv.org/abs/1504.08083.

R. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '14. IEEE Computer Society, Washington, DC, 2014, pp. 580–587. isbn: 978-1-4799-51185. doi: 10.1109/CVPR.2014.81.

I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook. org.

I. Goodfellow et al. "Generative adversarial nets". In: *Advances in Neural Information Processing Systems 27.* Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. url: http: //papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

S. Ha et al. "MarioNETte: Few-shot face reenactment preserving identity of unseen targets". *arXiv e-prints* (Nov. 2019), arXiv:1911.08139. arXiv: 1911.08139 cs.CV.

G. Hadjeres and F. Pachet. "DeepBach: A steerable model for bach chorales generation". *ArXiv e-prints* (Dec. 2016). arXiv: 1612.01010 cs.AI.

K. He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *The IEEE International Conference on Computer Vision (ICCV).* Dec. 2015.

N. Hegde et al. "Similar image search for histopathology: SMILY". *NPJ Digital Medicine* 2.1 (2019), p. 56.

S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. issn: 0899-7667. doi: 10.1162/neco.1997.9.8.1735. url: http://dx.doi.org/10. 1162/neco.1997.9.8.1735.

G. B. Huang E. Learned-Miller. *Labeled Faces in the Wild: Updates and New Reporting Procedures*. Tech. rep. UM-CS-2014-003. University of Massachusetts, Amherst, May 2014.

M. Hussain. “Security in connected cars”. In: *Proceedings of the European Automotive Congress EAEC-ESFA 2015*. Springer International Publishing, Cham, 2016, pp. 267–275. isbn: 978-3-31927276-4. doi: 10.1007/978-3-319-27276-4 24.

F. N. Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡1MB model size”. *arXiv* abs/1602.07360 (2016). url: http://arxiv.org/abs/1602.07360.

P. Isola et al. “Image-to-image translation with conditional adversarial networks”. *ArXiv e-prints* (Nov. 2016). arXiv: 1611.07004 cs.CV.

H. Jaeger. “The “echo state” approach to analysing and training recurrent neural networks with an Erratum note”. *German National Research Center for Information Technology* (2001). url: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf.

S. C. James, Yushan Zhang and Fearghal O’Donncha. “A machine learning framework to forecast wave conditions”. *arXiv preprint arXiv:1709.08725* (2017).

Y. Jia et al. “Direct speech-to-speech translation with a sequence-to-sequence model”. *CoRR* abs/1904.06037 (2019). arXiv: 1904.06037. url: http://arxiv.org/abs/1904.06037.

G. N. Kariniotakis, G. S. Stavrakakis and E. F. Nogaret. “Wind power forecasting using advanced neural networks models”. *IEEE transactions on Energy conversion* 11.4 (1996), pp. 762–767.

A. Karpathy and F-F Li. “Deep visual-semantic alignments for generating image descriptions”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015*, 2015, pp. 3128–3137. doi: 10.1109/CVPR.2015.7298932.

T. Kmet and M. Kmetova. “A 24H forecast of solar irradiance using echo state neural networks”. In: *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*. EANN ’15. ACM, Rhodes, Island, Greece, 2015, 6:1–6:5. isbn: 978-1-4503-3580-5. doi: 10.1145/2797143.2797166.

P. Korshunov and S. Marcel. “DeepFakes: A new threat to face recognition? Assessment and detection”. *CoRR* abs/1812.08685 (2018). arXiv: 1812.08685. url: http://arxiv.org/abs/1812. 08685.

W. Kristjanpoller, A. Fadic and M. C. Minutolo. “Volatility forecast using hybrid neural network models”. *Expert Systems with Appllication* 41.5 (Apr. 2014), pp. 2437–2442. issn: 0957-4174. doi: 10. 1016/j.eswa.2013.09.043.

Y. LeCun et al. “Generalization and network design strategies”. *Connectionism in Perspective* (1989), pp. 143–155.

C. Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. *arXiv* abs/1609.04802 (2016). url: http://arxiv.org/abs/1609.04802.

S. Levine, N. Wagener and P. Abbeel. “Learning contact-rich manipulation skills with guided policy search”. In: *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26–30 May, 2015*. 2015, pp. 156–163. doi: 10.1109/ICRA.2015.7138994.

S. Levine et al. “End-to-end training of deep visuomotor policies”. *ArXiv* abs/1504.00702 (2015). url: http://arxiv.org/abs/1504.00702.

S. Levine et al. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection”. *ArXiv* abs/1603.02199 (2016). url: http://arxiv.org/abs/1603.02199.

Y. Li. “Deep reinforcement learning: An overview”. *arXiv preprint arXiv:1701.07274* (2017).

C. Li, X. Guo and Q. Mei. “Deep memory networks for attitude identification”. *ArXiv e-prints* (Jan. 2017). arXiv: 1701.04189 cs.CL.

D. Li et al. “Reinforcement learning and deep learning based lateral control for autonomous driving”. *arXiv preprint arXiv:1810.12778* (2018).

X. Lu, D. Que and G. Cao. “Volatility forecast based on the hybrid artificial neural network and GARCH-type models”. *Procedia Computer Science* 91 (2016), pp. 1044–1049. Promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016).. issn: 1877-0509. doi:10.1016/j.procs.2016.07.145. url: http://www.sciencedirect.com/science/article/pii/ S1877050916313382.

E. Mansimov et al. “Generating images from captions with attention”. *CoRR* abs/1511.02793 (2015). url: http://arxiv.org/abs/1511.02793.

A. Marcu. “A local-global approach to semantic segmentation in aerial images”. *ArXiv* abs/1607.05620 (2016). url: http://arxiv.org/abs/1607.05620.

A. Mellit and A. Massi Pavan. “A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected {PV} plant at Trieste, Italy”. *Solar Energy* 84.5 (2010), pp. 807–821. issn: 0038-092X. doi: http://dx.doi.org/10.1016/j.solener. 2010.02.006. url: http://www.sciencedirect.com/science/article/pii/S0038092X10000782.

V. Mnih and G. E. Hinton. “Learning to detect roads in high-resolution aerial images”. In: *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part VI*. Ed. by Kostas Daniilidis, Petros Maragos and Nikos Paragios, Springer, Berlin, Heidelberg, 2010, pp. 210–223. isbn: 978-3-64215567-3. doi: 10.1007/978-3-642-15567-3 16. url: http://dx.doi.org/10.1007/978-3-642-15567-3 16.

S. A. Monfared and D. Enke. “Noise canceling in volatility forecasting using an adaptive neural network filter”. *Procedia Computer Science* 61 (2015), pp. 80–84. issn: 1877-0509. doi:10.1016/j.procs.2015.09.155. url: http://www.sciencedirect.com/science/article/ pii/S1877050915029853.

F. Monti et al. “Fake news detection on social media using geometric deep learning”. *rXiv preprint arXiv:1902.06673* (2019).

A. More and M.C. Deo. “Forecasting wind with neural networks”. *Marine Structures* 16.1 (2003), pp. 35–49.

C. Naik et al. “Contextual slot carryover for disparate schemas”. *arXiv preprint arXiv:1806.01773* (2018).

A. Owens et al. “Visually indicated sounds”. *ArXiv e-prints* abs/1512.08512 (2015). url: http://arxiv.org/abs/1512.08512.

X. B. Peng, G. Berseth and M. van de Panne. “Terrain-adaptive locomotion skills using deep reinforcement learning”. *ACM Transactions on Graphics (Proc. SIGGRAPH 2016)* 35.4 (2016).

C. Perlich et al. “Bid optimizing and inventory scoring in targeted online advertising”. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 804–812.

T. Poklepović, J. Arnerić and Z. Aljinović. “GARCH based artificial neural networks in forecasting conditional variance of stock returns”. *Croatian Operational Research Review* 5.2 (2014), pp. 329–343.

D. A. Pomerleau and D. A. Pomerleau. “ALVINN: An autonomous land vehicle in a neural network”. *Advances in Neural Information Processing Systems 1* (1989).

S. E. Reed et al. “Generative adversarial text to image synthesis”. *ArXiv e-prints* abs/1605.05396 (2016). url: http://arxiv.org/abs/1605.05396.

S. Ren et al. “Faster R-CNN: Towards real-time object detection with region proposal networks”. *arXiv* abs/1506.01497 (2015). url: http://arxiv.org/abs/1506.01497.

K. Rohrig and B. Lange. “Application of wind power prediction tools for power system operations”. In: *2006 IEEE Power Engineering Society General Meeting*. IEEE. 2006.

P. Romeu et al. “Time-series forecasting of indoor temperature using pre-trained deep neural networks”. In: *Proceedings of the 23rd International Conference on Artificial Neural Networks and Machine Learning – ICANN 2013 - Volume 8131*, Springer-Verlag New York, Inc., New York, NY, 2013, pp. 451–458. isbn: 978-3-642-40727-7. doi: 10.1007/978-3-642-40728-4 57.

A. Rössler et al. “FaceForensics++: Learning to detect manipulated facial images”. *CoRR* abs/1901.08971 (2019). arXiv: 1901.08971. url: http://arxiv.org/abs/1901.08971.

D. E. Rumelhart, G. E. Hinton and R. J. Williams. “Learning representations by back-propagating errors”. *Nature* 323 (Oct. 1986), pp. 533–536. doi: 10.1038/323533a0.

O. Russakovsky et al. “ImageNet large scale visual recognition challenge”. *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. doi: 10.1007/s11263-015-0816-y.

A. A. Rusu et al. “Sim-to-real robot learning from pixels with progressive nets”. *arXiv preprint arXiv:1610.04286* (2016).

A. E. L. Sallab et al. “Deep reinforcement learning framework for autonomous driving”. *Electronic Imaging* 2017.19 (2017), pp. 70–76.

G. Saon et al. “The IBM 2016 english conversational telephone speech recognition system”. *ArXiv e-prints* (Apr. 2016). arXiv: 1604.08242 cs.CL.

J. Schmidhuber. “Deep learning in neural networks: An overview”. *Neural Networks* 61 (2015), pp. 85–117. Published online 2014; based on TR arXiv:1404.7828 cs.NE. doi:10.1016/j.neunet.2014. 09.003.

F. Schroff, D. Kalenichenko and J. Philbin. “FaceNet: A unified embedding for face recognition and clustering”. *arXiv* abs/1503.03832 (2015). url: http://arxiv.org/abs/1503.03832.

K. Sharma et al. “Combating fake news: A survey on identification and mitigation techniques”. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.3 (2019), p. 21.

J. Shen et al. “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions”. *CoRR* abs/1712.05884 (2017). arXiv: 1712.05884. url: http://arxiv.org/abs/1712.05884.

D. Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. *Science* 362.6419 (2018), pp. 1140–1144.

D. Silver et al. “Mastering chess and shogi by self-play with a general reinforcement learning algorithm”. *arXiv preprint arXiv:1712.01815* (2017).

D. Silver et al. “Mastering the game of go with deep neural networks and tree search”. *Nature* 529.7587 (Jan. 2016), pp. 484–489. doi: 10.1038/nature16961.

A. J. R. Simpson, G. Roma and M. D. Plumbley. “Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network”. In: *Latent Variable Analysis and Signal Separation: 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25–28, 2015, Proceedings*. Ed. by Emmanuel Vincent et al. Springer International Publishing, Cham, 2015, pp. 429–436. isbn: 978-3-319-22482-4. doi: 10.1007/978-3-319-22482-4 50.

I. Sutskever, O. Vinyals and Q. V. Le. “Sequence to sequence learning with neural networks”. *ArXiv e-prints* (Sept. 2014). arXiv: 1409.3215 cs.CL.

Y. Taigman et al. “DeepFace: Closing the gap to human-level performance in face verification”. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR ’14. IEEE Computer Society, Washington, DC, 2014, pp. 1701–1708. isbn: 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.220.

A. Tampuu et al. “Multiagent cooperation and competition with deep reinforcement learning”. *ArXiv* abs/1511.08779 (2015). url: http://arxiv.org/abs/1511.08779.

M. Valdenegro-Toro. “Objectness scoring and detection proposals in forward-looking sonar images with convolutional neural networks”. In: *Artificial Neural Networks in Pattern Recognition: 7th IAPR TC3 Workshop, ANNPR 2016, Ulm, Germany, September 28–30, 2016, Proceedings*. Ed. by Friedhelm Schwenker et al. Springer International Publishing, Cham, 2016, pp. 209–219. isbn: 978-3319-46182-3. doi: 10.1007/978-3-319-46182-3 18.

A. van den Oord et al. “WaveNet: A generative model for raw audio”. *ArXiv eprints* abs/1609.03499 (2016). url: http://arxiv.org/abs/1609.03499.

H. Van Hasselt, A. Guez and D. Silver. “Deep reinforcement learning with double qlearning”. In: *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

P. Vincent et al. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion”. *Journal of Machine Learning. Research* 11 (Dec. 2010), pp. 3371–3408. issn: 1532-4435. url: http://dl.acm.org/citation.cfm?id=1756006.1953039.

O. Vinyals et al. “Show and tell: Lessons learned from the 2015 mscoco image captioning challenge”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

M. Wang and W. Deng. “Deep face recognition: A survey”. *arXiv preprint arXiv:1804.06655* (2018).

Y. Wu et al. “Google’s neural machine translation system: Bridging the gap between human and machine translation”. *ArXiv* abs/1609.08144 (2016). url: http://arxiv.org/abs/1609.08144.

W. Xiong et al. “Achieving human parity in conversational speech recognition”. *ArXiv e-prints* (Oct. 2016). arXiv: 1610.05256 cs.CL.

T. Young et al. “Recent trends in deep learning based natural language processing”. *IEEE Computational intelligenCe Magazine* 13.3 (2018), pp. 55–75.

R. L. Zeff and B. Aronson. *Advertising on the Internet*. John Wiley & Sons, Inc., 1999.

L. Zhang, S. Wang and B. Liu. “Deep learning for sentiment analysis: A survey”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1253.

J-Y Zhu et al. “Generative visual manipulation on the natural image manifold”. In: *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V*. Ed. by Bastian Leibe et al. Springer International Publishing, Cham, 2016, pp. 597–613. isbn: 978-3-319-46454-1. doi: 10.1007/978-3-319-46454-1 36.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com