Learning Idioms and Slang using a Deep Learning approach with Speech Recognition

Davide Berdin
{davide.berdin.0110}@student.uu.se

Master Thesis
Department of Information Technology

October 5, 2015

Abstract

To my family that never stopped believing in me

Acknowledgements

Contents

Chapter 1

Introduction

The pronunciation is one of the hardest part of learning a language among all the other componests, such as grammar rules and vocabulary. To achive a good level of pronunciation, non native speakers have to study and costantly practise the target language for a incredible amount of hours. In most cases, when students are learning a new language, the teacher is not a native speaker in which implies that the pronunciation may be influenced by the country where he or she comes from, since it is a normal consequence of second learning language [1]. In fact, [2] states that the advantages of having a native speaker as a teacher, lies in the superior liguistic compentences, especially the usage of the language more spontaniously in different communication situations. Pronunciation falls into those competences underlying a base problem in teaching pronunciation at school.

The basic questions we tried to answer in this work are:
1) why is pronunciation so important?
2) What are the most effective methods for improving the pronunciation?
3) What is the research state-of-art and what can we do to make it better?

The first question is fairly easy to answer. There are two reasons to claim why pronunciation is important: *(i)* it helps to acquire the target language faster and *(ii)* being understood. Regarding the first point, earlier a learner masters the basics of pronunciation, the faster it will become fluent. The reason is becuase *critical listening* with a particular focus on hearing the sounds will lead to gain fluency in speaking the language. The second point is **crucial** when working with other people, especially at these days where both in school and business the environment is often multicultural. Pronunciation mistakes may lead the person to being misunderstood affecting the results of a project for example.

With these statements in mind, [3] gives suggestions on how a lerner can effectively improve the pronunciation. Four important ways are depicted: *Conversation* is the most relevant approach to improve pronunciation, although, a supervision of an *expert guidance* that corrects the mistakes is foundamental during the process of learning. At the same time, learnes have to be pro-active to have conversation with other native speakers in such a way to costantly practising. *Repetitions* of pronunciation exercises is another important factor that will help the learner to be better in speaking. As last, *Critical listening*, that we also mentioned above, amplify the opportunity of learning the way native speakers pronounce words. In particular, for a learner is important to understand the difference when he or she is pronouncing a certain sentece with the one said by the native speaker. This method is very effective and is important for understanding the different sounds of the language and how a native speaker is able to reproduce them [4].
An important factor while learning a second language is to have a feedback about improvements. Teachers are usually responsible to judge the way the learners' progress. In fact, when teaching pronunciation, often it is used to draw the intonation and the stress of the words in such a way that the learner is able to see how the utterances should be pronounced. The *British Council* shows this practice [5]. The usage of visual feedbacks is the key of learning pronunciation and it is the main feature of this research.

In the computer science field, some works have been previously done regarding pronunciation. For instance, [6] helps lerners to acquire the tonal sound system of Mandarin Chinese through a mobile game. Another example is [7] in which the application provides a platform where learners of Chinese language can interact with native speakers and challenging them to a competition of pronunciations of chinese tones.

The idea behind this project is based on the fact that people need to keep practising their pronunciation to have a significant improvement as well as they need immediate feedbacks to understand if they are going in the right direction or not. The approach we used is based on these two factors and we designed the system to be as useful and portable as possible. The mobile application is where the user will test the pronunciation, a server through a machine learning technique will compute the similarity between the user's pronunciation and the native speaker's one and the results will be displayed on the phone.

We started collecting data from *American Native Speakers* in which we asked to pronounced a set of most used idioms and slangs. Each candidate had to repeat the same sentence four times trying to be as much consistent as possible. After we gathered the data, a preprocessing step is due since we are seeking for specific features such as voice-stress, accent, intonation and formants. This part is actually divided in two parts as well, where we used **PRAAT**[8] to extract the features from each audio file and then we used **MATLAB**[1] to average the four audio signals that each candidate said. After the avering step, there is a smoothing process in which we applied the *Exponential Filter* to each signal. A *resampling* was necessery since not all the signals have the same legth in time. The next step consisted in the so called **force alignment** where for each audio file we extract the *phonemes* that have been pronounced.
The same treatment is given to the audio signals given by the final user in order to compare the pronunciation.

The machine learning part is composed by a *deep neural network* trained using a statistical modelling method called **conditional random fields**. The system is trained using the features extracted during the data processing phase, coupled with the phonemes retrieved using the foce alignment. The phonemes are used as labels for the classification in which we indeed predict. To estimate the overall error between the native pronunciatio and the user, we use a method called **Word Error Rate** (WER), a common method metric for measuring the performance of a speech recognition system. In addition to *WER*, we used **Dynamic Time Warping** (DTW) to evaluate the similarity between features. These results will allow the user to have a better understanding of the differences for each characteristic.
DTW is not a part of the neural network, but it's a standalone algorithm.

When the server has computed the similarity, a score with the evaluation of voice-stress, accent, intonation and formants is given to the user. Those features are not the only one available. In fact, the application is capable to show the graph for each feature of the user's pronunciation and the one of the native speaker. In this way the user has a clear understanding of how he or she should **adjust** the way the utterance should be pronounced accordingly.

To measure the accuracy of our model, we asked 10 native speakers to rate the pronunciation of 10 non native speakers on 5 different sentences. The range of evaluation was between 0 and 10. From here we averaged all the rates and compared with the results obtained by the automatic system.

[1] http://www.mathworks.com

Chapter 2

Sounds of the General American English

In the *General American English* there are 41 different sounds in which can be structured by the way they are produced. In 2.1 is shown the kind of sounds with the respective number of possible productions. Each type will be described into a dedicated section of this thesis. An important factor is the way of how the *constriction* of the flow of air is made. In fact, to distiguish between *consonants*, *semivowels* and *vowels*, the *degree* of constriction is checked. Instead, for *sonorant* consonants the air flow is continuous with no pressure. *Nasal* consonants have an occlusive consonant made with a lowered velum allowing the airflow in the nasal cavity [**?**]. The *continuant* consonants are produced without blocking the airflow in the oral cavity.

Type	**Number**
Vowels	18
Fricatives	8
Stops	6
Nasals	3
Semivowels	4
Affricates	2
Aspirant	1

Table 2.1: Type of English sounds

2.1 Vowel production

Generally speaking, when a vowel is pronunced, there is no air-constriction in the flow. This means that the ariculators like the tongue, lips and the uvula do not touch allowing the flow of air from the lungs. The consonants instead have another pattern when producing them. Moreover, to produce each vowel, the mouth has to make a different shape in such a way that the resonance is different. 2.2 shows the way the mouth, the jaw and the lips are combined in a such a way to produce the acoustinc sound of a vowel.

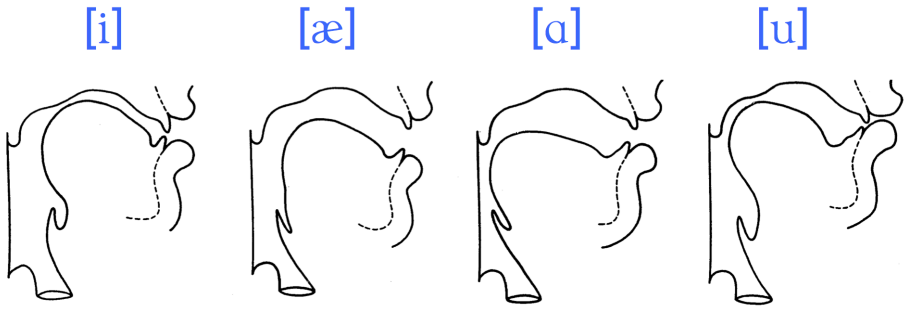

Figure 2.1: Vowels production

2.1.1 Vowel of American English

There are 18 different vowels in American English that can be grouped by three different sets: the **monopthongs**, the **diphthongs**, and the **schwa's** - or reduced vowels.

/iʸ/	iy	beat	/ɔ/	ao	bought	/ɑʸ/	ay	bite
/ɪ/	ih	bit	/ʌ/	ah	but	/ɔʸ/	oy	Boyd
/eʸ/	ey	bait	/oʷ/	ow	boat	/ɑʷ/	aw	bout
/ɛ/	eh	bet	/ʊ/	uh	book	[ə]	ax	about
/æ/	ae	bat	/u/	uw	boot	[ɨ]	ix	roses
/ɑ/	aa	Bob	/ɝ/	er	Bert	[ɚ]	axr	butter

Figure 2.2: Example of words depending on the group

The first column shows some examples of monopthongs. A *monopthong* is a clear vowel sound in which the utterance are fixed at both the beginning and at the end. The central part of the picture represents the dipthongs. A *dipthong* is the sound produced by two vowels when they occur within the same syllable [9]. In the last column are depictes some examples of reduced vowels. *Schwa's* refers to the vowel sound that stays in the mid-central of the word. In general, in the english language, the schwa is found in unstressed position [10].

2.1.2 Formants

A *formant* is the resonant frequency of a vocal track that resonate the loudest. In a spectrum graph, formants are represented by the peaks. In 2.3 it is possibile to see how the three first formants are defined by the peaks. The pictures is the *envelope* of a spectogram of the vowel **[i]**. Frequencies are the most relevant information to determine which vowel has been pronounced. In general, within a spectrum graph there may be a different number of formants, although the most relevant are the first three and they are named **F1**, **F2** and **F3**.

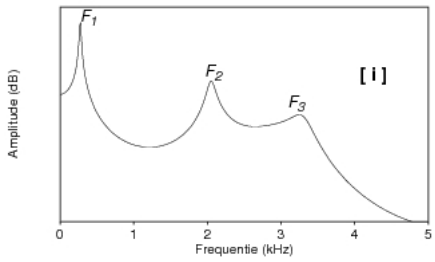

Figure 2.3: Spectral envelope of the [i] vowel pronunciation. F1, F2 and F3 are the first 3 formants [11]

Chapter 3

Speech Analysis

3.1 Prosody

3.1.1 Pitch Tracking

3.1.2 Discrete Logarithmic Fourier Transform

3.2 Local Tones vs. Global Intonation

3.2.1 Voice Intensity

3.2.2 Voice Stress

Formants

Chapter 4

Artificial Neural Networks

4.1 Artificial Neuron

Our brain is composed by biological neurons and an artificial neuron (or AN) is a representation of it. Every AN can gather signals from other nurons or from the environment, and after an elaboration it transmits another signal to all the other ANs that are connected to it [12]. A rapresentation of AN is depicted in 4.1.
Each connection to the artificial neuron has a numerical weigth associated to it in which the input signal is hold back. The value of the weight can be either positive or negative. In most cases, the sums of each node are weighted and then given as input to a *non-linear* function called **transfer function** or **activation function** [13]. The activate function defines the output value of the node and typically, the *Step Function*, *Sigmoid Function* and a *Softmax Function* are the most used.

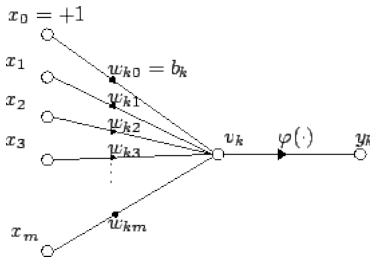

Figure 4.1: Representation of an Artificial Neuron [13]

From the mathematical point of view, we can define an artificial neuron as follow:
given $m + 1$ inputs with signals from x_1 to x_m and weights values from w_0 to w_m. The *bias* is then defined by the input x_0 in which a value of 1 will be assigned. The bias value allows us to *shift* the curve of the activation function to a certain direction and it is defined with $w_{k0} = b_k$ [13].
The output of the AN is:

$$y_k = \varphi\left(\sum_{j=0}^{m} w_{kj} x_j\right)$$

4.2 Network Function

When there are many aritificial neurons interconnected between each other in the different layers, we form a *network*. 4.2 shows an example of ANN where the **inputs** are represented by the first layer in which they send data through the connection to the second group of neuron. The connection between two neuros is called *synapses* where the **weigth** is stored. The second layer is connected to the third one that represents the **output** of the network. There can be multiple stratums between the inputs and the outputs and these are called *hidden layers*.

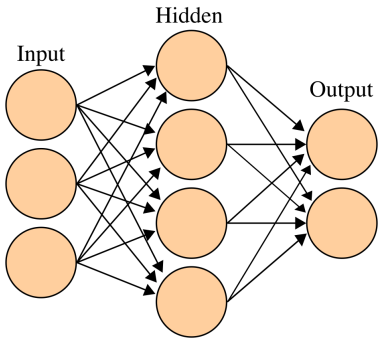

Figure 4.2: Example of ANN [14]

Typically, a neural network is defined by three factors:

1 How the different layers are interconnected

2 How the the weights are updated (learning process)

3 How the neuron's input value is converted to its output activation (activation function)

4.3 Learning

For every application the design of a neural network is different and after it has structured, the NN is ready to be trained. The first step is the initialization of the weigths. Normally they are initialized with random values, although, in [15] they developed a method to *"determining the optimal initial weights of feedforward neural networks based on the Cauchy's inequality and a linear algebraic method"*. More, in [16] they tested seven different weights' initialization for twelve different problems. Thus, even the initialization of the connections values is application-dependant.
The learning paradigms can be grouped in three major categories: **supervised**, **unsupervised** and **reinforcement learning**.

Supervised Learning

This learning technique is the task of inferring a function from labeled training data [17] [18]. The set responsible for training the model is composed by *training examples* in which every sample consists of an *input object* and a **desidered** *output value*. What the algorithms does is to analyze the train dataset and produce an inferred function that will be used to map the new examples [17]. Basically, the algorithm can be seen as *learning* with a *teacher*, in the sense that the there is costantly a feedback on the status of the application.

Unsupervised Learning

While in supervised learning, the system has a desidered output given from the training dataset, in the unsupervised paradigm, the system has to learn to estimate the right output given a new input [19]. In *classification*, this output is a class label whereas in *regression* is a real number. There are several ways to model this kind of learning system. *Clustering*, using *Self-Organizing Maps*, *K-means* or *hierachical clustering* are among the most famous approaches.

Reinforcement Learning

In reinforcement learning, the system takes actions to interact with its own environment. Each action will affect the state of the environment in which will produe a result in a form of either *reward* or *punishment*. The goal of this learning paradigm is to learn which is the best sequence of actions that maximises the rewards or minimises the punishments. There are quite a few approaches to find the best sequence of actions. The most famouse are *Monte Carlo methods*, *Temporal Difference methods* and *Direct Policy search methods*.

4.4 Multilayer Perceptron

The model created by the Multilayer perceptron (MLP) serves to map a set of inputs onto a group of ouputs. The main feature of the MLP is that it a *feedforward* artificial neural network. The MLP is formed by a defined number or layers in which every layer is *fully connected* to following one. Every neuron of the network has a nonlinear activation function, with the exception of the input nodes. The techinque used in the MLP is of the kind of supervised and it uses the **backpropagation** for training the neurons [20]. Backpropagation is discussed more in details in 4.4.

Learning through Backpropagation

The learning phase in the neural network occurs in the moment that the connection weigths change based on the error value in the output. The error is calculated comparing the result the network produced with the expected one. This is a typical example of supervised learning because the network compares the result it just obtained with the one that it was expecting. This process is done through **backproagation**.

In backpropagation, the error output of the node j in the nth sample of the training dataset is given by

$$e_j(n) = d_j(n) - y_j(n) \tag{4.1}$$

where d is the expected output whereas y is the output value obtained from the neuron. The weights are then adjusted in such a way that the error is minimized. With 4.2 we are able to determine the corrections to apply given an output value produced by a perceptron.

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{4.2}$$

At this point using 4.3 we are able to determine the amount of change for each weight. This is done by **gradient descent** which is a first-order optimization algorithm. Gradient descent is used to find *local minima* of a function, where *"it takes steps proportionally to the negative of the gradient of the function in that point"*[21]. The opposite instead, it means that it is approaching to the *local maxima* of the function. ALthough, in this way, the process would be called *gradient ascent*[21].

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_j(n) \tag{4.3}$$

In 4.3, η represents the *learning rate* whereas y_i is the output of the previous neuron [20]. The learning rate parameter is one of the most importan parameters when design a neural network. The reason is that, the value used for it ensures that the weights are converging as fast as possible avoiding waivings.

The calculation of the derivates depends on the field v_j where this value changes itself. Continuing, 4.3 can be simplified to 4.4 where ϕ' represents the *derivate* of the activation function described before. Note that this does not changes itself.

$$\frac{\partial \varepsilon(n)}{\partial v_j(n)} = e_j(n)\phi'(v_j(n)) \tag{4.4}$$

4.5 Deep Learning

Deep Learning has several definitions in which those have been changing in last 10 years. Definition number 5 reported in [22] says the following: *"Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text."*.
There are two main aspects of the level of representations described above:

1) a model consists of several layers of nonlinear processing units

2) for supervised and unsupervised approaches the feature representation have more abstract layers

It is possible to classify Deep Learning as an intersection between different research areas, such as: neural networks, pattern recognition, signal processing, etc..[22]

4.5.1 Deep Neural Networs

As described in 4.5, a *deep neural network* is composed by several hidden layers of perceptrons between the input(s) and the output(s) [23]. As for the artificial neural networks, DNNs are able to model complex non-linear relationships. [24]
Adding extra layers to a ANN permits to each layer to *specialize* in solving a certain problem. For example, in *visual pattern recognition* the neurons in the first layer can try to learn how to recognize edges in a picture, whereas those in the second layer might focus in learning ho to recognize more complex shapes like a square or a triangle built up from the previous edges. The next layers will try to learn even more complex shapes. In the case of speech recognition, we could split the problem in different parts as well. The first layer could focus in recognizing phonemes, the second layer can learn about the pitch voice, and so on.
The usage of multiple hidden layers gives to DNNs some advantages in learning how to solve complex problems compared to the shallow networks.[25] An example of DNN is depicted in 4.3.

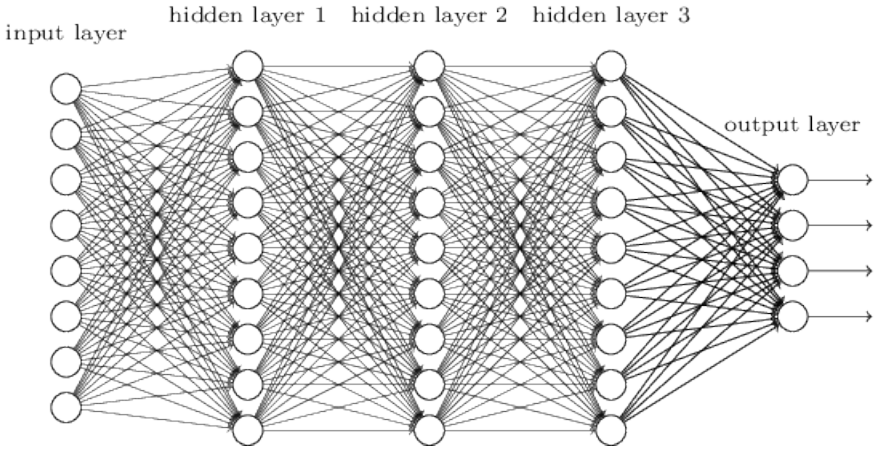

Figure 4.3: Example of Deep Neural Network [25]

Architecture

Deep Neural Networks are oftenly designed as feedforward networks and trained with the *back propagation* algorithm. Although, to train acoustinc models for Automatic Speech Recognition (ASR) *Convolutional Deep Neural Networks* (CNNs) presented better results than the typical feedforward design. Despite the similarity between the two networks, in CNNs *"the neurons in the inputs of hidden units in layer m are from a subset of units in layer m-1, units that have spatially contiguous receptive fields"*. [26]
To update the weights the *stochastic gradient descent* is used as follow:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}} \tag{4.5}$$

where η is the *learning rate* whereas C is the *cost function*. The activation function and the learning type have a direct influence in the choice of the cost function. For instance, a typical choice of the activation function for a supervised learning approach on a problem of multiclass classification are either **softmax function** or **cross entropy function**. The mathematical definition of *softmax function* is defined in 4.6 where p_j represents the class probability whereas x_k and x_j are the total input to neuron x and j of that layer. *Cross entropy function* is defined by 4.7 where d_j is the target probability of the output neurons j and p_j is the probability output for j after applying the activation function. [24]

$$p_j = \frac{exp(x_j)}{\sum_k exp(x_k)} \tag{4.6}$$

$$C = -\sum_j d_j log(p_j) \tag{4.7}$$

Chapter 5

Implementation

5.1 Data Collection

5.2 PRAAT

5.3 Training the Neural Network

5.4 The Application

5.5 Setting up the Server

Chapter 6

Evaluation

Chapter 7

Conclusions

Chapter 8

Future Works

Bibliography

[1] T. M. Derwing and M. J. Munro, "Second language accent and pronunciation teaching: A research-based approach," *Tesol Quarterly*, pp. 379–397, 2005.

[2] P. Medgyes, "When the teacher is a non-native speaker," *Teaching English as a second or foreign language*, vol. 3, pp. 429–442, 2001.

[3] A. Gilakjani, S. Ahmadi, and M. Ahmadi, "Why is pronunciation so difficult to learn?," *English Language Teaching*, vol. 4, no. 3, p. p74, 2011.

[4] M. Rost and C. Candlin, *Listening in language learning*. Routledge, 2014.

[5] "Word stress - british council," 2015. accessed 2015-09-28. Available: https://www.teachingenglish.org.uk/article/word-stress.

[6] D. Edge, K.-Y. Cheng, M. Whitney, Y. Qian, Z. Yan, and F. Soong, "Tip tap tones: mobile microtraining of mandarin sounds," in *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 427–430, ACM, 2012.

[7] A. Head, Y. Xu, and J. Wang, "Tonewars: Connecting language learners and native speakers through collaborative mobile games," in *Intelligent Tutoring Systems*, pp. 368–377, Springer, 2014.

[8] P. Boersma and D. Weenink, "{P} raat: doing phonetics by computer," 2010.

[9] "Dipthong," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Diphthong.

[10] "Dipthong," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Schwa.

[11] "The spectrum of acousting," 2015. accessed 2015-09-28. Available: http://www.hum.uu.nl/uilots/lab/courseware/phonetics/basics_of_acoustics_2/formants.html.

[12] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

[13] "Artificial neuron," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Artificial_neuron.

[14] "Artificial neural network," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Artificial_neural_network.

[15] J. Y. Yam and T. W. Chow, "A weight initialization method for improving training speed in feedforward neural network," *Neurocomputing*, vol. 30, no. 1, pp. 219–232, 2000.

[16] M. Fernandez-Redondo and C. Hernandez-Espinosa, "Weight initialization methods for multilayer feedforward.," in *ESANN*, pp. 119–124, 2001.

[17] "Supervise learning," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Supervised_learning.

[18] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[19] Z. Ghahramani, "Unsupervised learning," in *Advanced Lectures on Machine Learning*, pp. 72–112, Springer, 2004.

[20] "Multilayer perceptron (mlp)," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Multilayer_perceptron.

[21] "Gradient descent," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Gradient_descent.

[22] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[23] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[24] "Deep learning," 2015. accessed 2015-09-08. Available: https://en.wikipedia.org/wiki/Deep_learning.

[25] "Why are deep neural networks hard to train?," 2015. accessed 2015-09-08. Available: http://neuralnetworksanddeeplearning.com/chap5.html.

[26] "Convolutional neural networks (lenet)," 2015. accessed 2015-09-08. Available: http://deeplearning.net/tutorial/lenet.html.