Final Project – Hybrid Transfer Learning Tutorial

**IDS-705 Principles of Machine Learning**

*Yuan Feng, Sebastián Soriano Pérez, Vishaal Venkatesh,*

*Abhiraj Vinnakota, Roderick Whang*

**Abstract** – This tutorial explains the concepts and motivations behind transfer learning and seeks to provide an example of the technique used in practice. When tackling a machine learning problem, we may face a situation in which we do not have enough training data in the domain we are interested in. However, there is sufficient data available in a related domain. Transfer learning takes advantage of the available data in a related domain to train a model for our target domain. Transfer learning has been shown to improve performance and reduce the costs of acquiring more labeled data. This tutorial will focus on applying transfer learning to deep neural networks for image classification problems, using a feature extraction (embedding) approach. We will first explain the methods and formal concepts behind transfer learning, and then provide an example to measure and understand its performance.

1. **Introduction**

[Provide a description concept/algorithm, motivate your reader as to why he/she should care about this question. How has the technique been used in practice and what are some motivating examples of its use?]

Quantum computing has been extensively featured in the news recently. While the field itself has made considerable progress since the 90s, we are still a long way from having quantum computers that can perform useful tasks. A useful task in this context means any computing process that answers a specific question. Before we dive into the current state of affairs in quantum computing, we need to understand the basics of quantum computing as they are considerably different from that of a classical computer. The fundamental unit of quantum information is the quantum bit or the qubit. This is analogous to the bit in a classical computer. A qubit can exist in either the zero state |0⟩ , the one state |1⟩ or in a linear combination of the above. The notation used above is known as a bra-ket notation first popularized by the English physicist Paul Dirac. A vector in the ‘ket’ notation is represented as |*v*⟩ and is simply a column vector. In contrast, a vector in the ‘bra’ notation is represented as ⟨*v*| which is simply a row vector. The following equation represents the |0⟩ and |1⟩ in their equivalent column vector representations (Equation 1).

|  |  |
| --- | --- |
|  | (1) |
|  |  |

These qubits can also exist as a linear combination of and which is depicted as follows (Equation 2). This is a unique property of qubits and is known as the superposition principle.

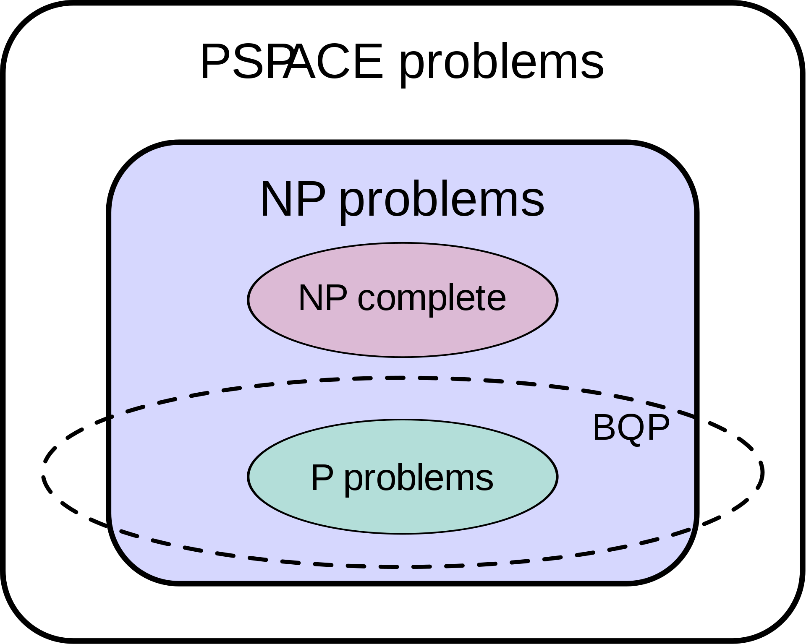
|  |  |
| --- | --- |
|  | (2) |

Here, and are known as the probability amplitudes associated with the qubit existing in the . Without going into the mathematical intricacies associated, it will also be stated here that the actual probability of finding the qubit in the state is 2 and state is 2. In other words, the following equation (Equation 3) will always hold true.

|  |  |
| --- | --- |
|  | (3) |

The probabilistic nature coupled with the superposition principle of qubits is in stark contrast to the deterministic nature of bits. However, it has to be stated here that the probabilistic nature of qubits only holds until the time the qubits are measured. In other words, the moment an external entity wishes to know the value of the qubit, the probabilistic nature quickly disappears, and the qubit assumes the value of either or depending on the values of and .

Next, it is important to understand why quantum computers may have added computational powers and may be able to offer a certain quantum-speedup in certain types of problems. We shall be able to understand this by comparing how bits and qubits operate. For example, consider the scenario where we have access to 50 bits and 50 qubits. Because each bit can either take the value of 0 or 1, the total number of different combinations in which the system can exist in is 250. However, at any given point of time, only 1 combination out of 250 can exist. In the case of qubits, however, all 250 simultaneously exist in superposition as a linear combination. This parallel processing ability is what gives quantum computers the advantage. It has to be mentioned here that while the above explanation is lucid, it is also vastly simplistic and even mildly erroneous. If parallel processing was indeed the reason why quantum computers have promising applications, then all classical algorithms must have a comparable quantum algorithm with a verifiable quantum speedup. This, however, is not the case and most classical problems cannot be solved faster by a quantum computer. There are a handful of problems where quantum computers offer an exponential speedup relative to their classical counterparts. The hope is that more such problems will be identified in the future with advancements in quantum information science (Figure 1).



**Figure 1.** *P refers to the set of all problems that can solved by a classical computer in polynomial time. NP refers to the set of all such problems for which given a solution, it can be verified in polynomial time. Finally, BQP refers to all such problems that can be solved by a quantum computer in polynomial time. Note that . The hope is that we find more problems in the set*

*Also, while still under debate, we have assumed*

If a quantum computer does not entirely rely on parallel processing to produce speedups, how does a quantum computer work then. According to the fundamental of quantum physics, if all possible solutions to a given problem were super positioned as a linear combination of each other, then each solution would be just as likely as any other. In other words, this is not how a quantum computer works. In contrast, a sequence of carefully chosen quantum gates (similar to logic gates in classical computing) are applied on the qubits. These gates can be viewed as simple matrix multiplications on super positioned qubits. The result of the application of such gates is that most trivial, unimportant results will collapse (similar to waves destructively interfering or simply cancelling out). The remaining states that have non-zero amplitudes and hence non-zero probabilities (these are akin to waves constructively interfering and adding up) hopefully contain the solution to our problem. We will look at one example of a quantum gate to get an idea. This is the Hadamard gate as given in Equation (4).

|  |  |
| --- | --- |
|  | (4) |

The Hadamard gate when applied to the and qubit yields the following (Equations 5 & 6). The results are also known as the Hadamard-mapped states.

|  |  |
| --- | --- |
|  | (5) |

|  |  |
| --- | --- |
|  | (6) |

Note that squaring the coefficients of the resulting states into which the Hadamard gates maps yields 0.5 in both cases (Squaring ). In other words, the application of the Hadamard gate creates superposition of the qubits wherein both the and the have a 0.5 probability of existing. Simply put, if a user where to measure a Hadamard mapped qubit there is a 50% chance the result will be and a 50% chance the qubit will be Other quantum gates used in this literature have been included in the appendix.

The application of quantum computing to the classification of images is not expected to provide a quantum speedup. However, we wish to develop a quantum circuit and use it on top of the ResNet18 architecture to classify images. This is at the cutting-edge topic of research in the field of Hybrid-Transfer learning The goal of this study is to evaluate the performance of the quantum classifier trained on top of ResNet18 and compare its performance to its classical counterpart.

1. **Background**

[This section should cite problems that have been previously addressed that relate to your work, and the key takeaways of the studies that explored that work. The idea here is to place the problem you’re working on in context and to let the reader know that you’re not working in a knowledge vacuum. For finding relevant literature, a good starting point is Google Scholar.]

1. **Methods**

In order to understand transfer learning, we must define two important concepts first: *domain* and *task*. In a machine learning, the **domain** consists of the feature space in the dataset, with features or predictors (), and its marginal probability distribution [1]. The marginal probability distribution is the probability distribution of each of the variables with no reference to the values of the rest of the variables. A **task** consists of the label space and an objective predictive function , which can be used to predict the class label and corresponds to the conditional probability distribution learned from the training data (where ) [1]. For a graphical representation see Figure 4.1.

Transfer learning is done by first training a machine learning model with training data having a source domain and a source task . For a binary classification problem, the training dataset would consist of pairs of values, where , the feature vector is , and . It is assumed that there is sufficient training data available to train the machine learning model, and this data is usually balanced. Then, transfer learning utilizes the information learned from this model and applies it to train new model with data having a specific target domain and a target task . Transfer learning aims to improve the learning of the target predictive function using the knowledge acquired by training the source model, when or [1]. Therefore, it is possible to try to estimate the target conditional probability distribution by using the information found through the training of the source model. See Figure 4.2.

The transfer learning techniques can be applied to different scenarios, depending on whether it is the target and source domains, or tasks that differ. The domains differ when the source and target feature spaces are different () or when the source and target marginal probability distributions are different (). Likewise, the tasks differ when the source and target label spaces are different (), when the conditional probability distributions are different (), or both. There is a wide range of transfer learning techniques that could be applied in any of these situations.

**Inductive transfer learning** is applied when the source and target tasks are different, regardless if their respective domains are the same or not. **Transductive transfer learning** is used when the source and target tasks are the same, but their respective domains are different. Finally, **unsupervised transfer learning** is applied to problems where the target task is different and focuses on unsupervised learning problems such as clustering or dimensionality reduction. We will focus on and provide examples of inductive transfer learning techniques from now on, particularly when there is plenty of labeled data in the source domain and the focus is to improve performance on a smaller set of labeled data available on the target domain.

Transfer learning is particularly popular for training neural networks for image classification, especially for convolutional neural networks. When we need to train a classification model but there is limited amount of labeled data available, we can use the information obtained from training a different model with more reliable data in a related domain. Image classification problems is an instance where the application of inductive transfer learning methods (when the source and target marginal probability distributions differ) is particularly useful.

A neural network for image classification consists of an input layer where each neuron corresponds to one of the features (where ), a varying number of hidden layers, and an output layer where each neuron corresponds to one of the labels of . The general idea behind how a neural network “learns” is that each layer after the input layer captures slightly more complex features in the images. The first layers’ neurons activate when there are specific edges in certain regions of the images, and subsequent layers start detecting more complex patterns and shapes. The final layer receives as an input more easily identifiable and complex features, so that it is easier to correctly detect which features are found on each label of .

A common approach to applying transfer learning for computer vision is to utilize a previously trained model on data with the same feature space but different marginal probability distribution, when the labeled data is sufficient. Then, the final layers of the pre-trained neural network are “removed”, and the remaining layers and their weights are fixed and used as feature extraction models for the new data in the target domain and task. The final layers of the model are then retrained again on the target data. This process is shown in Figure 4.3. This approach is known as **feature extraction** or **embedding**. This method improves the training time and model performance. In the following section we will apply the feature extraction technique as an example.

1. **Examples of the technique in practice**

[Apply your technique to sample problems to demonstrate the technique. Clearly describe the application and evaluate performance. Illustrate both strengths and weaknesses. Comparisons to related techniques are strongly encouraged.]

Ants vs. Bees

Imagen que contiene foto, interior, diferente, tabla

Descripción generada automáticamente

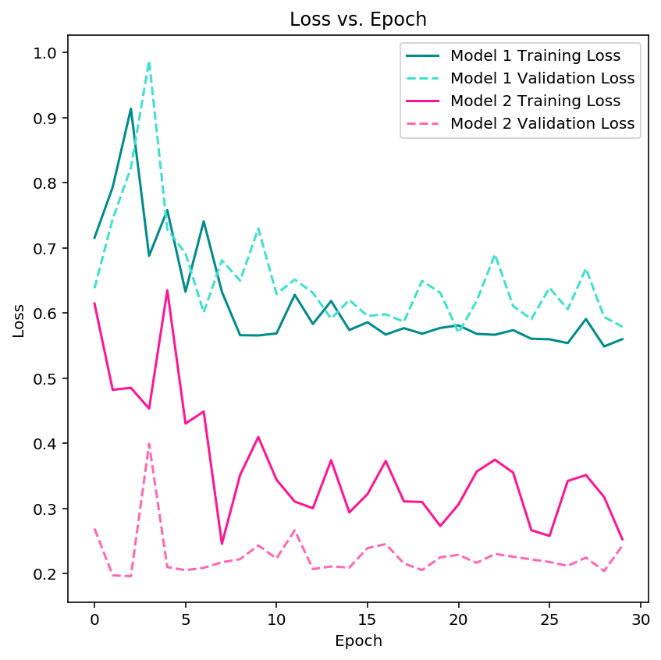


Imagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto, mapa

Descripción generada automáticamente

Dogs vs. Cats

Perro mirando hacia arriba

Descripción generada automáticamente

Imagen que contiene texto

Descripción generada automáticamente

Imagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto

Descripción generada automáticamente

1. **Summary**

[Summary of the method, its uses, as well as its strengths and weaknesses as compared to other similar techniques.]

1. **Roles**

[Since this is a team project, we want to know what your specific contribution was to this project. Provide detail on your individual role and how it contributed to the competition. Each team member should clearly articulate an individual role.]

**Yuan Feng**:

**Sebastián Soriano Pérez**: Researched and authored Background and Methods sections. Implemented non quantum models.

**Vishaal Venkatesh**:

**Abhiraj Vinnakota**:

**Roderick Whang**:

1. **References** [no word limits]

[An alphabetical list of references cited in this work. A minimum of 10 are required. Consider using the Zotero citation manager for collecting and compiling your references.]