Evaluating the Ability of Machine Learning Models to Learn Generalizations of Contextual and Temporal Tasks

Titus Lungu (tituslungu@gmail.com)
University of California, Los Angeles
Henry Samueli School of Engineering and Applied Science

Abstract

The ability of machine learning models to generalize tasks and learn concepts without contextual reliance is crucial for increases in speed, efficiency, and overall performance of intelligent systems. When humans learn a concept, they are able to isolate it from its context almost immediately and then generalize to many other situations where the concept might be found. The evaluation of several machine learning models' ability to learn certain tennis swings within a specific context and generalize to another context is presented. Failure and success cases are discussed as well and compared to human performance in similar scenarios. The code, data, and a demo video can be found at www.tituslungu.com/tennisai.

Keywords: Machine learning generalization; learning temporal tasks; tennis training; experience for generalization; context learning.

Introduction

Humans are exceptionally apt at not only learning information well and quickly, but also at generalizing that information. When a task is performed or an item is observed, people are able to efficiently extract critical features to aid recognition and/or repetition. Beyond extracting features, humans are likely able to isolate parts of a temporal task or an observed object in such a way that a change in surrounding context of that isolated piece will still allow for a correct interpretation to be made.

This ability could have two underlying reasons to it. On the one hand, parts of a recognition task are isolated such that they are not conjoined with the context in which they are initially observed, allowing for broad generalization. This might occur through a mechanism akin to clustering where statistically varying items are understood to be different from one another and are grouped as such. When that item is then learned, or labeled, it is done so without having specific context directly tied to it. Therefore, next time the item is observed, it can be more easily identified regardless of context.

A second, and likely more critical reason for the ability of humans to truly *understand* and learn concepts at a more abstract level is due to direct experience with those concepts. Living in the same world ever since conception, the human mind has a deeply rooted, even intrinsic understanding of the concepts with which it comes into contact on a regular basis. More so, it understands the context surrounding those concepts so well, it in fact makes it easy to decouple context from concept, allowing generalization of knowledge.

Both of these notions seem to be implied to a degree by Minsky (1975), where he theorizes his framework for representing knowledge, focusing particularly on vision. Minsky explains his theory that the human mind likely remembers commonly seen objects, essentially isolating different items, as explained above. Then, when those items come up again, the experience gained from having seen them repeatedly before helps ease the computational load required to process the seen image and to identify specific items in it. Again, generalization is seen to be performed by isolating items on their own and gaining a deeper understanding of them through experience.

State-of-the-art machine learning and AI algorithms have a much harder time with this notion of generalization. When generalization is achieved, enormous amounts of data are required to train the model. This is because, during the training, the model will "learn" everything about each particular training example. This includes context, noise, and other artifacts. With a lot of data on one particular item or task to recognize, the features of the relevant item will eventually rise to the top, and all the surrounding noise will be ignored as their is no relationship between the noise throughout the different training examples. Evaluating the ability of machine learning models to learn information in context and then generalize to out of context, compared to human ability to do so, is essential in improving the performance, efficiency, and speed at which these models can learn.

Methodology

The goal was to study the ability of current machine learning and AI techniques to learn knowledge versus generalize it, and compare that with humans' respective abilities. In an attempt to do so with as little ambiguity as possible, it was desirable to use a test scenario that provided as much separability between the abilities of learning and generalizing. Classification of actions during tennis play was chosen because different swings are easily distinguishable but still allow for a high degree of variability during play.

Data was collected from five tennis players using a custom logging app on an Android Wear Smartwatch borrowed from the UCLA Center for Smart Health. The tennis players were of different skill level, though all around the broader intermediate level, and data for each swing type from all the players was grouped. These two factors (grouping data from several players and using players of varying skill level) were incorporated in hopes of allowing for a degree of generalization in learning.

The smartwatch was strapped to the tennis player's wrist, and all data was for swings performed with the left hand only (chosen based on the tennis players available for the study). The watch provided data about swings from a myriad of internal sensors, further discussed in a subsequent section. In order to ensure a useful study, four classes were chosen for classification: forehand swing, backhand swing, overhand serve, and a neutral stance (waiting for the ball).

Swing data was collected in two phases. First, snippets of data for each class were collected for training the machine learning models and testing their ability to learn. Next, a sequence of swings was recorded as would occur in a match, which was used for testing the ability of the models to generalize.

Data Collection

Snippet Data

Each training example is collected as a "snippet". The tennis player presses the "start recording" button on the smartwatch, performs one of three hits (forehand, backhand, or overhand), then presses "stop recording" on the watch. Each such snippet was around four to six seconds in length. Around 170 training snippets of each hit were collected between all five tennis players.

For the neutral phase, two non-hitting sequences were recorded (several minutes in length each) where the tennis player moved around the court as if in anticipation of receiving a ball to return. These sequences were later split into smaller snippets similar to those recorded for the hit classes by scanning the signal with a window of some length, moving along the signal by a specified stride.

Match Sequence Data

For evaluating the ability to generalize, the generalized test should be "similar" to the training data from which the model learned, while possessing some alteration. This "similarity" is loosely defined, but generally implies that the test task is the same as the training task only within a different context, from a different perspective, with additional noise, surrounded and clouded by other tasks, etc.

Since the training data was collected in single, stationary snippets, the generalized test data was collected as a sequence from a match. Hits were performed in arbitrary order with neutral phases in-between. The tennis player moved around the court to return hits, as occurs in a match. As with the neutral training data, the match sequence was all one recording rather than snippets. In order to test on the sequence later, it was scanned with a window of specified length moving along the signal by a specified stride. These windows were then fed to the model for prediction.

Preprocessing and Feature Selection

Python3 was used for preprocessing the data and training and testing the models. In preprocessing the data for training, two major considerations were required: choosing the sensors whose data to use for training, and choosing the features to extract from that data. It was found that the optimal sensor data to use was that from the accelerometer and gyroscope. Figures 1-5 show, left to right, the accelerometer and gyroscope signals of one training example from each of the four classes, as well as from a window of the match sequence testing data. Very distinct patterns can be seen between the different classes.

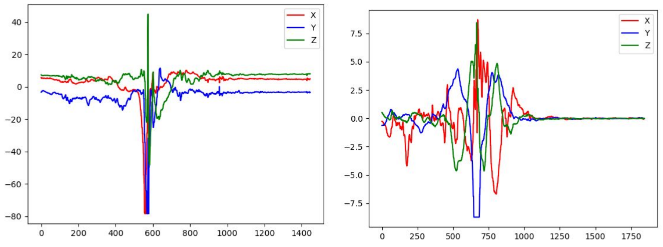

Figure 1: Backhand signal.

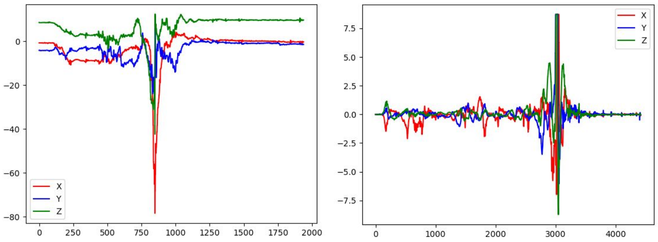

Figure 2: Forehand signal.

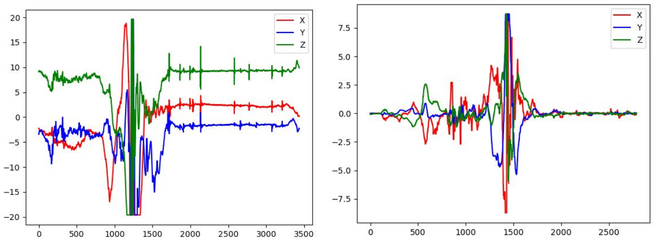

Figure 3: Overhand serve signal.

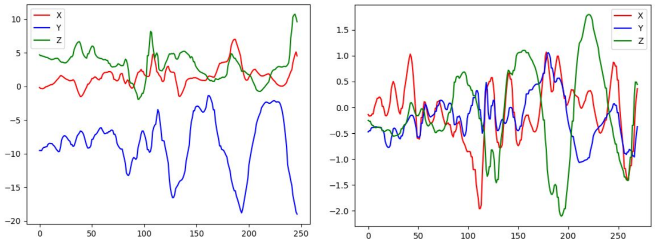

Figure 4: Neutral stance signal.

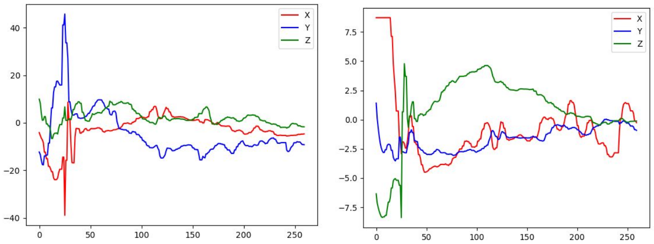

Figure 5: Match sequence test window signal.

The features that proved to provide the most separability between classes and therefore the highest learning performance were extracted by transforming the signals to frequency domain. Using a Fast Fourier Transform, the magnitude of the amplitudes of the five most dominant frequencies were used as features, as well as the standard deviation of the frequencies of the top 200 dominant frequencies. The power magnitudes of the five most dominant frequencies were also extracted from the Power Spectral Density of the signal, as well as the respective standard deviation. Finally, a discrete wavelet transform was performed on the signal and the same features were extracted as with the FFT and PSD. In the time domain, the norm of the signal was computed and the maximum magnitude observed as well as its standard deviation were used as features. After considering all three positional axes (for the frequency domain features) and the signals from both sensors used, each training example was represented by a total of 112 features. Figure 6 shows the frequency transformations of the forehand gyroscope signal shown in Figure 2.

In order to ensure that features across training examples were on the same scale, each of the 112 features were normalized across all training examples. This drastically improved prediction accuracy, by about 30-40%. Labels were assigned to the data as one-hot encoded vectors. Since the training examples have a period of irrelevant motion at the beginning and end (pressing “start” and “stop” on the watch), the signal was cropped by varying amounts in order to improve prediction accuracy.

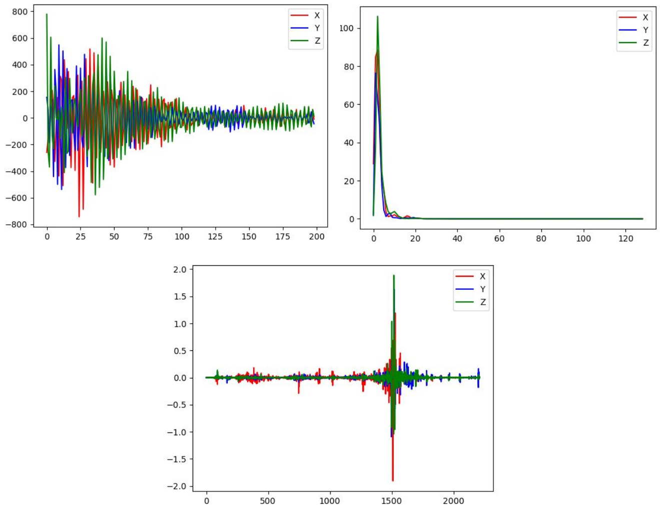

Figure 6: FFT, PSD, and wavelet transforms (left to right, top down) of forehand gyroscope signal shown in Figure 2.

Model Creations

Several machine learning models were implemented in order to compare learning performance across the models. K-fold cross validation was used with an optimal K value of 10 in order to cycle through different training and testing samples for a more robust measure of performance. Scikit-learn was used to implement five models, specifically, K-Nearest Neighbors, Decision Tree, Gaussian Naive Bayes, Support Vector Machine (with a linear kernel, a regularization constant of 1, and a kernel hyperparameter of 0.7), and Random Forest (with 20 estimators).

A sixth model, a deep neural network, was implemented in TensorFlow. The optimal architecture was found to be a network with 4 hidden layers containing 50, 30, 20, and 10 nodes respectively, all with ReLU activations. The activation on the four nodes of the output layer was a softmax function in order to obtain results as the probability of an example belonging each class. Softmax cross entropy was used as the cost function and an Adam optimizer for training the weights, with an initial learning rate of 0.001 and running for 400 epochs. The Adam optimizer was chosen because of it's implementation of learning rate decay. This is preferred because, as training progresses, it often becomes increasingly difficult to improve the model due to the fact that taking large steps in gradient descent is more likely overshoot the minimum and increase the cost the closer the model gets to a local minimum. The Adam Optimizer also implements momentum to help reach a global minimum rather than a local minimum. Momentum uses the velocity of the descent to optimally increase the learning rate and overcome small hills and continue to a better minimum.

Many other techniques exist as well to prevent a network from overfitting the training data, which is of key importance in the network's ability to generalize. Three such techniques were implemented, namely dropout, L2 regularization, and mini-batching. However, due to the relative small size of the network they did not improve prediction accuracy and even decreased the accuracy in some cases.

Dropout is a technique that randomly sets some node activations to zero and is useful in large networks because it prevents the model from relying too heavily on any one piece of information, causing it to learn redundant representations which aides the ability of the model to generalize. L2 regularization penalizes the network for updating weights too drastically. A drastic change in the weights may likely signify the network relying too heavily on one or several features, which could mean it is memorizing the training data (ie: overfitting). Mini-batching divides the training data into batches that are iteratively fed to the network to train during every epoch, causing the network to see different examples successively.

Results

Snippets

Each model was first trained on the snippet data and tested using K-fold cross validation. Training and validation accuracies when training on all four classes using the two sensors mentioned and all 112 features are shown in Table 2. All accuracies are in the 90th percentile range, with the deep neural network performing the best. While all the training accuracies are higher than the respective validation accuracies, most of the models do not seem to overfit the data. However, in the case of the decision tree and random forest classifiers, the validation accuracies are much smaller than the training accuracies, implying that the model may have in fact memorized the training data (which is reinforced by the fact that the training accuracy is 100% in both cases). This is not much of a surprise however due to the nature of these models, which do in fact learn a more rigid representation of the classes from the training data. Random forests use multiple decision trees to prevent overfitting, resulting in the 3% increase in validation accuracy.

The perceived similarity between different classes was also analyzed using the neural network by running the model on different combinations of two classes at a time. Table 2 shows the validation accuracy when only two classes are used. The higher the accuracy, the less similar the model perceives those two classes to be. While it seems that the neutral phase is perceived to be the most different from any other class, the other pairwise groupings are not off by much.

It makes logical sense that neutral is the most different when paired with any other class, since the data is very stagnant compared to the sensor readings of the other classes when the ball is hit. Surprisingly however, the forehand swing seems to be more similar to the backhand swing than the overhand serve, which is not expected. Due to forehand and overhand both being hits in the same direction, ones might expect them to be much more similar than, say, forehand and backhand, since in this case the swing is in performed in different directions. This may be caused by the fact that more force is applied during an overhand serve as compared to a normal forehand swing, whereas both forehand and backhand swings apply a similar amount of force to the ball. The fact that most of the features used were in terms of magnitude and therefore non-directional may contribute to this result as well.

Table 1: Performances of different models for learning tennis actions.

Model	Training Accuracy	Validation Accuracy
Neural Network	99%	97%
KNN	95%	94%
Decision Tree	100%	90%
Gaussian Naive Bayes	94%	93%
SVM	92%	91%
Random Forest	100%	93%

Match Sequence

The performance of the model on the match sequence was significantly less accurate than on the training snippets. As expected, the model was not able to generalize the learned class representations when the context of the hits changed. During data collection of the snippets, the player is stationary. However, during the match, the player is constantly moving to receive and return the ball. More so than simply increased movement, the intensity of play is much higher during a match. For example, when a forehand swing is performed, the player first pulls back before swinging the racket. This "pulling back" motion may very well be confused for a backhand swing. Depending on the intensity and height of the forehand swing, it could also be confused with an overhand serve. The neutral state is very noisy, and as seen in Figure 5, the match sequence is rather noisy as well, so the neutral state acts as a kind of default.

Match sequence performance is generally higher when swings are performed more similarly to the training snippets, particularly when the player is calm and less mobile. The gentler the hit, the better the accuracy. Opposite conditions are foreign to the model and that is when the classification fails. Different windowing parameters on the test sequence affect the performance as well. The next section goes into greater detail regarding the effect of contextual data on the performance of the model on the match sequence.

Table 2: Accuracy of neural network when only running model on two classes, implying the perceived similarity of the two classes.

	Fore-hand	Back-hand	Over-hand	Neutral
Fore-hand		96%	98%	100%
Back-hand			100%	100%
Over-hand				100%
Neutral				

Effects of Data Cropping on Contextual Understanding and the Ability to Generalize

An interesting observation is made regarding how cropping the training data affected the accuracy of the model's prediction on the snippet data versus the match sequence. When the data is uncropped, having artifacts on both ends of the snippet due to the nature of starting and stopping recording manually while collecting the data. These artifacts are not part of the hits that are being classified, however, they do provide the context in which the hit was performed. However this context is very similar surrounding all the swings, so removing it by cropping the data would seem beneficial in decreasing the perceived similarity between different classes during the training. However, when cropping about 30% of the data off both ends of the snippets, the validation accuracy decreases to 92-94%. This decrease is likely caused by the fact that the hit in each recording occurs at varying times and cropping the data likely removes some of the signal from the actual hit as well.

A far more interesting observation is regarding the performance of the model on the match sequence when cropping the training data. As mentioned, the excess data in the training examples makes up the context of the hits. Since the training and match sequence have very different contexts, it is preferential to remove the training context when wishing to predict on the match sequence. This in fact was found to be empirically true. When the training data was not cropped, the model predicted the neutral class for almost all of the match sequence data. This is not extremely surprising as the majority of the signal in the training snippets is essentially "neutral", or non-action. That mixed with the short impulse from the hit likely caused confusion for the model when it saw the match sequence where a lot movement occurred continuously. As shown in Figure 5, the signals during the match sequence have less variability in them than do the signals in the training data because of the player's continuous motion during play. Signals like those of the large impulses during a hit in the training data are not as noticeable in the match sequence. Instead, the match sequence actually looks much more similar to the data from the neutral class in terms of homogeneity.

When the data was cropped by 30% on both end of the snippets, predictions on the match sequence noticeably improved. The accuracy was slightly higher, and class predictions were updated much more often while progressing through the sequence. By removing the context from the snippets, the model was able to focus only on the hits themselves. Then, when seeing the new data during the match, it was at least able to understand that different hits were occurring because it had learned a model that provided some isolation of each class from any noise or other irrelevant data.

A video with examples of correct classification and incorrect classification on the match sequence may be viewed at www.tituslungu.com/tennisai. The majority of the correctly classified hits have a similar context surrounding them as the training snippets. The video shows performance for several windowing parameters as well as several different amounts of front- and back-end data cropping.

Discussion and Analysis

These results have interesting comparisons and contrasts with human performance in learning and generalization of contextual and temporal tasks, both specifically in tennis as well as on a broader level. Since the specific task analyzed for this study was playing tennis, the most relevant comparisons will be made to how human tennis players learn and generalize playing the sport.

In (Scott, 1998), tennis players' performance was measured based on the ability to return balls and how that ability was affected by receiving video training. Similarly to previous such studies, participants were shown videos of game events and ask to predict the outcome or make a decision as to how they would react. Previous studies have found that players are able to learn to react and make decision faster from video training, but that this does not necessarily transfer to playing the game in reality. This would seem to agree with the notion of the importance of experience in order to overcome contextual differences. Watching a video of an event is significantly different from participating in the event. Similarly, video training in previous studies mentioned in the paper showed to decrease decision time but not increase the accuracy. This too seems indicative of the importance of experience in learning. Seeing an event in a game is a visual task, whether it is a live game or a video. However, the matter of *which* decision is made is affected by many more factors such as the player's physical presence, the progress of the match, the importance of the match, how the player feels, physical and mental exhaustion, etc.

In the study in (Scott, 1998) however, video training seems to have in fact increased the performance of the tennis players. A cited reason is that perhaps this is due to the players receiving more training in this study than in previous ones. The incorporation of feedback is likely a major factor as well. The tennis players in the study were shown a video of a tennis player preparing to serve a ball, and the video was shown just up to right before the ball makes contact with the racket. The player watching the video was then asked to predict what type of serve would be played. The videos were played at different speeds, starting with the slowest speeds, and the players could only advance to the next speed when an accuracy threshold was reached for the performance. This feedback likely allowed the players to learn the context and semantics of this new task of responding to truncated videos of tennis plays. Initially, all players performed poorly because the context of the task (it being a video as well as the video being truncated) is very contextually different than actually playing tennis. More specifically, this contextual difference is difficult to overcome because their is little experience to based on as it is uncommon to watch a video with the intent of reacting, as if the person watching it was there. That is an experience that only occurs in the physical world, and is well-known there.

In (Barnett, 2008), the authors analyze different tennis swings and which are preferable for different players based on their track records, and then make observations and recommendations as to how a player might perform more optimally. This indication that a tennis player may be able to change their pattern of play should simply be a matter of combining different swings. If a tennis player learns specific swings, like the deep neural network model learned, but unlike the model is able to generalize, then playing a different sequence of swings is simply a matter of organizing "isolated" items into a different context (ie: the new sequence).

Such an ability to combine and order learned items is an interesting notion that is dependent on the complexity of the system being built from said items. Tennis has a short list of simple rules, so reordering swings is logically easy. However, a more complex task might be organizing furniture in a room. This is itself an abstract and undefined ordering of items that requires more computational resources to execute properly.

In addition, however, reordering tennis swings to create a new generalization from the swings might in fact be more difficult for the human to *execute*. While a tennis player and generalize and understand the advantage of reordering their swings and can make a plan to do so, execution is much less explicitly thought of and much more a habitual task based on style and repetition. Therefore, humans do not require relearning of concepts given different contexts, but they do require retraining once those concepts or tasks have to executed. In contrast, if a machine decides or is told that a different course of action is advisable, execution is quick and direct. The impediment for the machine, as has been discussed, is reaching the conclusion in the first place. Consider a scenario where a machine automatically collects plays and outcomes (scores, win/loss, etc.) and continues to learn what the preferred sequence of swings is from every new match. If after a long time of a certain set of sequences performing well, suddenly for some reason a different kind of sequence becomes better, it could take some time for weights to be properly adjusted for the machine to learn this. In contrast, a human can detect a critical change like this quickly and understand what new actions should be taken. Based on this, the duration of implementation is likely a function of habits and/or age for humans and a function of the weighting on learned actions for a machine or model.

A way to overcome this obstacle in the machine model would be by using reinforcement learning, where the machine incurs losses for poor performance and rewards for correct performance. This helps shift weights much faster and the model will converge to playing the new optimal sequence of swings sooner. This is similar to how the human brain handles this situation, as it has a notion of loss and reward which causes it to notice required changes in the model faster. Interestingly, (Pollard, 2008) finds that there is usually a quadratic relationship between the risk taken on the serve and the probability of winning the play. In other words, the riskier a player’s serve is, the more likely they are to win the play when the serve is completely successfully. This would indicate a high reward in a reinforcement model in the human mind when planning a new play.

Conclusion

Human generalization is a key element in the ability to understand concepts regardless of context and to operate effectively and efficiency in the world. Through such methods as concept isolation, experience, and reinforcement learning humans are able to create a more profound and meaningful representation of the their in their minds, directly aiding their ability to generalize. Evaluating a deep neural network with high success in learning contextual task provided insights into when the model was able to generalize the knowledge and learn the task out of context, and when it was not. A further survey of similar studies involving human performance of tennis seem to reinforce the reasoning derived for an ability or lack thereof to generalize.

References

Minsky, M. (1975). A Framework for Representing Knowledge. *The Psychology of Computer Vision*. P. Winston (Ed.), McGraw-Hill.

Scott, D. et al. (1998). Training Anticipation for Intermediate Tennis Players. *Behavior Modification*. Volume 22 (pp. 243-261).

Barnett, T., Meyer, D., Pollard, G. (2008). Applying Match Statistics to Increase Serving Performance. *Med Sci Tennis.* (pp. 24-27).

Pollard, G. (2008). What is the Best Serving Strategy? *Med Sci Tennis.* (pp. 34-38).