A neural network model of hierarchical category development

Chris Gorman

August 24, 2015

Introduce self, say something like "My presentation is about our model of how the brain develops a category hierarchy." Adjust based on previous presentations

1. Plenty of computational models exist which are able to classify and categorize objects
2. These models, however, don't quite explain how people actually *develop* these category representations, so that was our goal going into this
3. In cognitive psychology, the prevailing idea is that we develop a hierarchy of categories throughout our lives. This hierarchy is split into three levels.

Postgraduate Symposium 2015

A neural network model of hierarchical category development

Chris Gorman

August 24, 2015

- Basic-level

1. The basic-level is, essentially, what you and I refer to when discussing an object. What are you sitting in right now? A chair. What is in front of you? A desk. What do you drive? Maybe a car, or a motorcycle. Those are all basic-level categories.
2. The basic-level provides a very cognitively efficient representation of objects. Things in the same basic-level category share many similarities and have very few differences.
3. These categories emerge because category members share frequent co-occurrences of properties. The prototypical dog has four legs, fur, a tail, etc. These features aren't randomly distributed, so we pick up on ones that frequently appear together and develop categories from them.
4. The categories are also defined functionally. You sit on a chair but you don't necessarily sit on furniture.

Hierarchical Categorization

- Basic-level

- Basic-level
- Superordinate-level (Liu et al., 2001)

1. Above the basic level we have the superordinate level which holds categories like furniture or vehicle. At this level, categories contain a lot of intra-category differences, both with properties and functionality
2. For example, a car and a boat are both vehicles, but they have a huge number of differences, ranging from shape, to size, to operation and beyond.
3. Also, superordinate-level categories actually vary culture to culture. For example, our two superordinate categories of animals and plants are separated into three categories for certain Australian aborigines: biological, food, and totemic (each human is thought to have a spiritual connection or a kinship with another physical being, such as an animal or plant, often called a spirit-being or totem.)

Hierarchical Categorization

- Basic-level
- Superordinate-level (Liu et al., 2001)

- Basic-level
- Superordinate-level (Liu et al., 2001)
- Subordinate-level

1. On the opposite end of the spectrum, below the basic-level, lies subordinate categories, such as sedan, dentist's chair, etc. They are defined, in part, by what makes them different from the prototypical member of their basic-level category. A pug is a dog which is short and has a wrinkly face.
2. Functionally, what you are able to do with a member of a subordinate-level category isn't much different than what you could do with other members of its basic-level category. For example, what you can do with a recliner is not much different than what you can do with an office chair.
3. Finally, subordinate-level categories are also defined by the subtle correlations of their features. These are the correlations that are essentially overwhelmed by the stronger ones which define their basic-level category.

Hierarchical Categorization

- Basic-level
- Superordinate-level (Liu et al., 2001)
- Subordinate-level

Idea
Basic-level categories emerge from strong correlations of features (Rosch et al., 1976).

1. I'll say this for emphasis. The key take-away here is that basic-level categories emerge from strong correlations of features. That being the case... (next slide)

Hierarchical Categorization

Idea

Basic-level categories emerge from strong correlations of features (Rosch et al., 1976).

How do we learn the subtle correlations of subordinate-level categories?

1. (say it) how do we learn the subtle correlations of subordinate-level categories? If these subtle correlations are overwhelmed by the stronger ones which make up the basic-level category, then... (next slide)

Hierarchical Categorization

- **How do we learn the subtle correlations of subordinate-level categories?**

How do we learn the subtle correlations of subordinate-level categories?
Corollary
Learning subordinate-level categories requires a different mechanism than the one used for the basic-level.

1. We must offer another explanation for the emergence of subordinate-level categories

Hierarchical Categorization

- **How do we learn the subtle correlations of subordinate-level categories?**

Corollary

Learning subordinate-level categories requires a different mechanism than the one used for the basic-level.

1. To that end, we developed the Dominant Property Assembly Network, or DPAN for short. DPAN is a neural network model which receives a series of input vectors representing token objects and learns both basic-level and subordinate-level categorical representations of the data.

Hierarchical Categorization

- **How do we learn the subtle correlations of subordinate-level categories?**

Corollary

Learning subordinate-level categories requires a different mechanism than the one used for the basic-level.

Based on this information, we've developed the **Dominant Property Assembly Network** (DPAN) to model the acquisition of basic and subordinate-level categories.

- **How do we learn the subtle correlations of subordinate-level categories?**

Corollary

Learning subordinate-level categories requires a different mechanism than the one used for the basic-level.

Based on this information, we've developed the **Dominant Property Assembly Network** (DPAN) to model the acquisition of basic and subordinate-level categories.

Figure: Layout of the input items

1. The input data we work with is laid out as such. We use a bit vector to represent a token object such that each bit represents an abstract, arbitrary feature of that object. For example, one bit could mean "short legs" or "squished nose" or something. The main idea is that each vector represents one of eight different cat or dog breeds based on the combination of features. Every dog breed has bits 0 through 9 active, every cat breed has bits 6 through 15 active, with the overlap representing features shared by both dogs and cats. After that we have a series of bits representing the different breeds, where two bits indicate features specific to that breed. Finally, at the end, we have a set of 5 random idiosyncratic properties which are simply unique, completely uncorrelated, random features.

Input Data

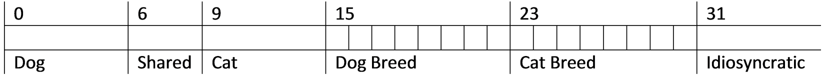
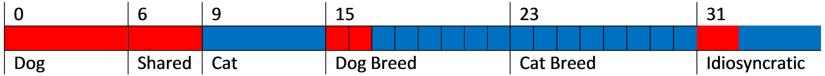


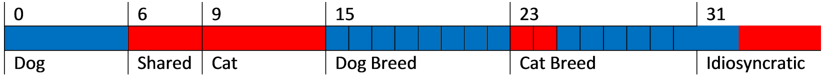
Figure: Layout of the input items

Sample Input: Pug



1. This is an example of an input vector representing pug. As you can see, the dog features are all active, the cat features are not, and there are two bits active which represent "pug-like" features, unique to this breed.

Sample Input: Tabby



1. And this is an example of a tabby vector. Again, notice the cat properties are active, dog properties are inactive, and the two "tabby features" are active.

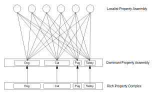


Figure: DPAN is organized into three layers. The **rich property complex** (RPC), the **dominant property assembly** (DPA) and the **localist property assembly** (LPA)

1. This is the architecture of DPAN. Starting from the bottom, the rich property complex holds the raw input vector. Above that, the dominant property assembly provides a sort of "property workspace" by holding a volatile copy of the rich property complex. This plays a very important role when learning subordinate-level categories, which I will discuss shortly. Next is a series of synaptic weights which connect the DPA to the localist property assembly. In the localist property assembly, each unit encodes an internal representation of the network's learned categories.

The Dominant Property Assembly Network - Overview

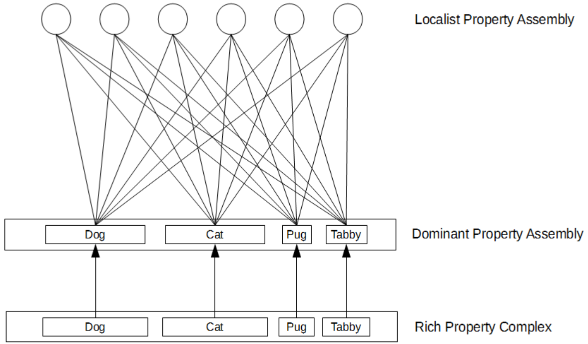


Figure: DPAN is organized into three layers. The **rich property complex** (RPC), the **dominant property assembly** (DPA) and the **localist property assembly** (LPA)

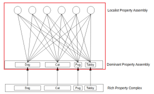


Figure: CPCA (O'Reilly & Munakata, 2000) provides the core learning mechanism for DPAN

1. DPAN's learning is accomplished using O'Reilly and Munakata's Conditional Principal Component Analysis, or CPCA, algorithm. During training, an input vector is read into the rich property complex. It is then copied into the dominant property assembly. The network calculates the activity of the units in the localist property assembly, chooses the one with the highest output, and selects it as the winner. It then updates the weights from the winning unit to the DPA

Conditional Principal Component Analysis

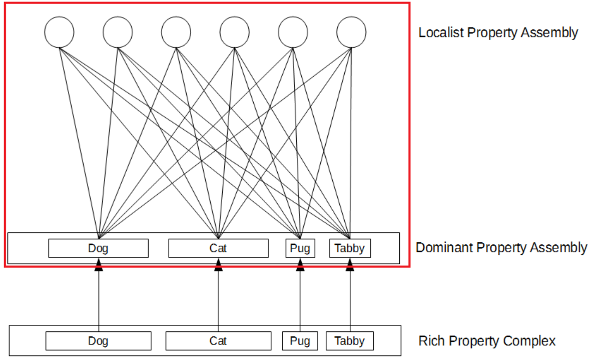


Figure: CPCA (O'Reilly & Munakata, 2000) provides the core learning mechanism for DPAN

$\vec{y} = \mathbf{W}^T \vec{x}$ (1)

1. These are the important equations used by the CPCA learning algorithm in DPAN. The activity of each unit is the weighted sum of the input vector.

Conditional Principal Component Analysis

$$\vec{y} = \mathbf{W}^T \vec{x} \tag{1}$$

$$\vec{y} = \mathbf{W}^T \vec{x} \quad (1)$$
$$\Delta w_{ij} = \alpha_j y_j (x_i - w_{ij}) \quad (2)$$

1. The weights of the winning unit are updated using the hebbian learning rule equation 2, where alpha is the learning rate, y is the unit's output, and x is the input. Essentially, this rule dictates that if the unit is active and the input is on, the weight grows stronger. If the unit is active and the input is off, the connection weakens. It also enforces a floor and ceiling on the value of the weights, in our case 0 and 1 respectively. If the unit is not active, nothing happens.

Conditional Principal Component Analysis

$$\vec{y} = \mathbf{W}^T \vec{x} \tag{1}$$

$$\Delta w_{ij} = \alpha_j y_j (x_i - w_{ij}) \tag{2}$$

$$\vec{y} = \mathbf{W}^T \vec{x} \quad (1)$$
$$\Delta w_{ij} = \alpha_j y_j (x_i - w_{ij}) \quad (2)$$
$$w_{ij} = P(y_j = 1 | x_i = 1) \quad (3)$$

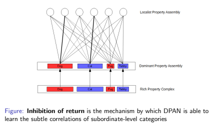
1. Oreilly and munakata provide a derivation proving that the value of any given weight is just the probability that the unit's output is one given that the input is 1. Thus, CPCA is able to represent correlations in input data.
2. Recall that basic-level categories are developed by observing the strong correlations of features, but we need another mechanism to learn the subtle correlations which are overwhelmed by the strong ones. This being the case, cpca in-and-of-itself is not sufficient to learn subordinate-level categories. To provide a mechanism which learns subordinate-level categories, DPAN utilizes Inhibition of Return, or IOR.

Conditional Principal Component Analysis

$$\vec{y} = \mathbf{W}^T \vec{x} \tag{1}$$

$$\Delta w_{ij} = \alpha_j y_j (x_i - w_{ij}) \tag{2}$$

$$w_{ij} = P(y_j = 1 | x_i = 1) \tag{3}$$



IOR allows DPAN to investigate the differences between a token object and its basic-level category in order to learn their more subtle correlations. To illustrate this, I'll walk through the steps the network takes during training. At this state, there is an input vector for "pug" sitting in both the RPC and the DPA. This network has already seen several inputs, so it already has a unit which encodes the basic-level category "dog."

Inhibition of Return

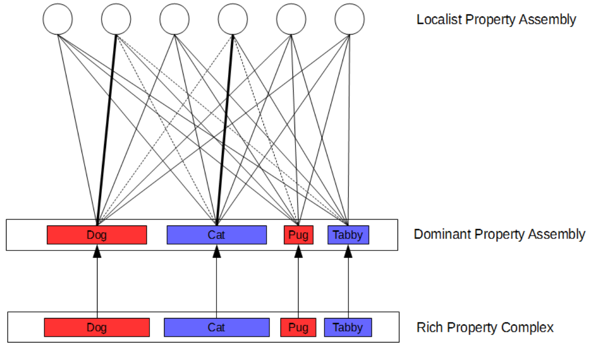
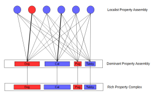
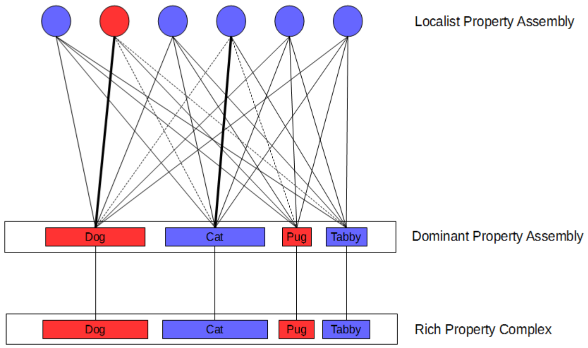


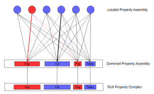
Figure: **Inhibition of return** is the mechanism by which DPAN is able to learn the subtle correlations of subordinate-level categories



The activity of each unit is calculated and a winner is selected, shown here in red. Since the unit for “dog” most closely represents the given input, that is selected to be the winning unit.

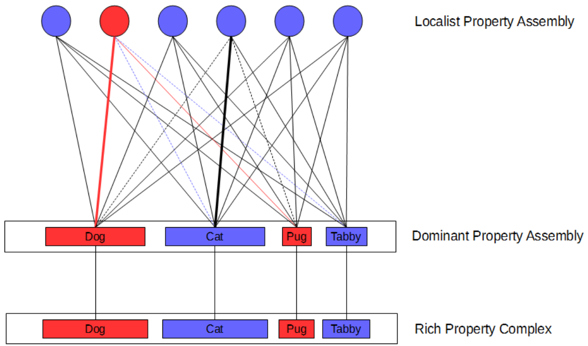
Inhibition of Return

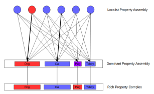




Next, the weights from the DPA to the "dog" unit are updated. Again, everything up to and including this point are all part of the standard CPCA algorithm.

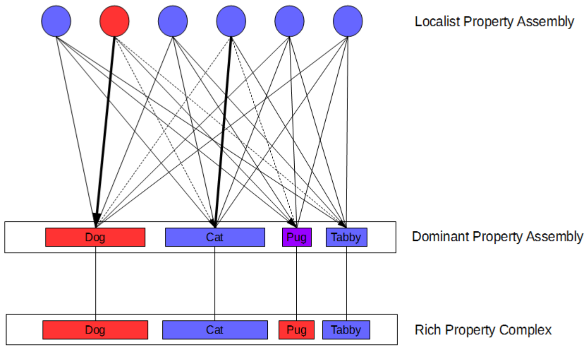
Inhibition of Return

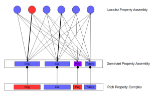




Now the network begins the IOR steps. The value of the winning unit's weight is copied into the DPA. That is, the DPA now contains the network's internal representation of the basic-level "dog" category.

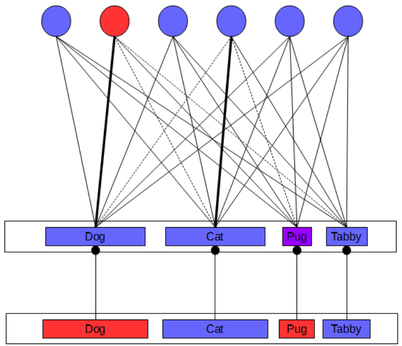
Inhibition of Return

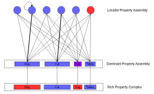




In order to investigate the difference between the basic-level category and the input object, the DPA is subtracted from the RPC. That is, the network's basic-level category is subtracted from the observed input object. What's leftover are the subtle correlations which are not fully represented in the network's basic-level category. In this case, it would be the "pug-like" features. Since all pugs are dogs and share the same dog-like features, those are wiped-out during this property-level inhibition.

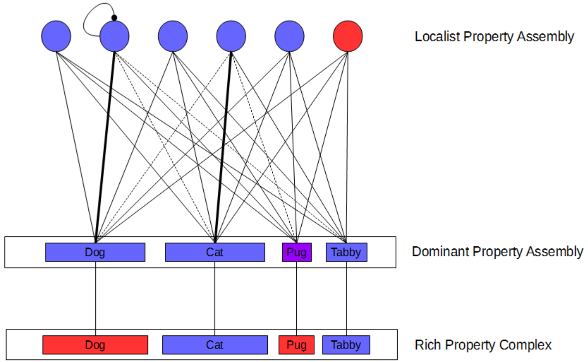
Inhibition of Return

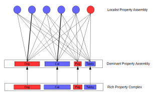




Now that the DPA basically only contains the interesting "pug-like" features, the activity is recalculated and a new winner is selected. This is the second inhibition operation: the network does not allow the same unit to win twice, since doing so would obviously just have the basic-level category win again.

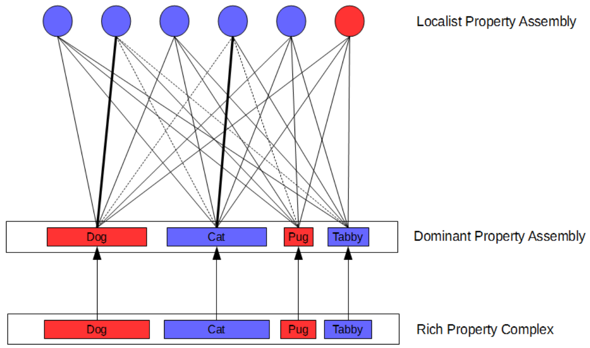
Inhibition of Return

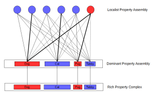




WIth a new winner selected that best represents the pug-like features in the DPA, the network next copies the RPC back into the DPA. This is because the winning unit needs to learn ALL of the features of the input object, not just the subtle ones.

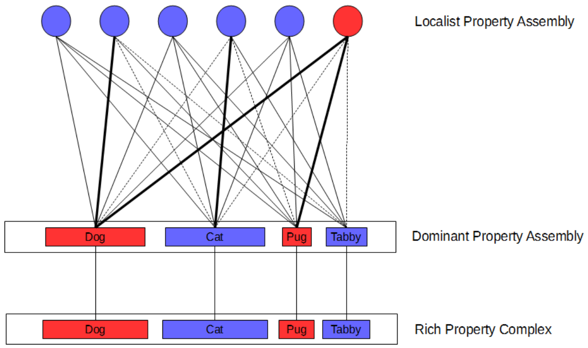
Inhibition of Return





Finally, the weights of the new winner are updated. So what we have here is a winning unit which really likes these pug features, but then updates its weights based on the both the pug features as well as the dog features.

Inhibition of Return



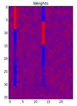


Figure: Execution after 200 training items

1. Up next we have the actual results of a DPAN execution. The network was presented with 5000 input vectors with a uniform probability of being a dog breed, either pug, spaniel, beagle, or corgi, or a cat breed, either tabby, maine coon, siamese, or persian.

Results

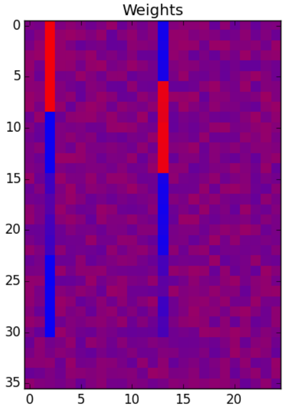


Figure: Execution after 200 training items

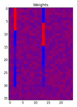


Figure: Execution after 200 training items

1. In these examples, I'm showing a heat map of the weight matrix. Each column vector represents one unit in the LPA.

Results

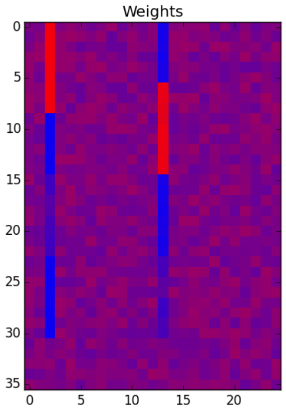


Figure: Execution after 200 training items

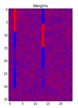


Figure: Execution after 200 training items

1. In the first phase of training, the network forms basic-level categories. When presented with one of the dog breeds, the network will select the unit representing dog and update the weights accordingly. This is analogous to a person seeing, say, a pug and saying "That is a dog."

Results

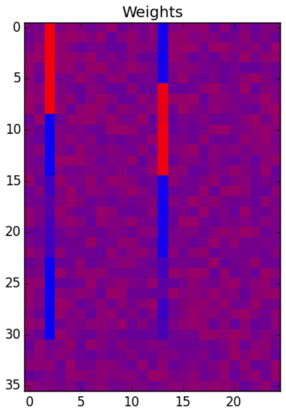


Figure: Execution after 200 training items

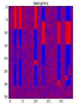


Figure: Execution after 1600 training items

In the next phase of training, the basic-level categories have been solidified and the network begins to use its IOR operation to learn subordinate-level categories. For most of the subordinate-level categories, the basic-level unit still wins first, then the inhibiton operation allows the subordiante-level unit to win next. This phase of training is analogous to a person seeing a pug and saying "That dog is a pug."

Results

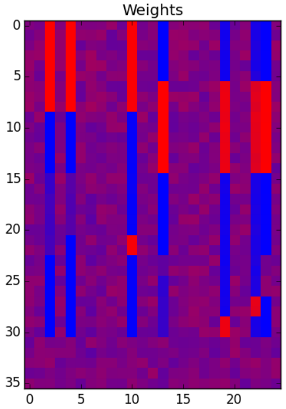


Figure: Execution after 1600 training items

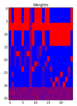


Figure: Execution after 5 epochs (25000 training items total)

In the third phase of training, the units have settled and all subordinate-level categories have at least one representative localist unit. When presented with a token object, the winning unit is that of the subordinate-level category, effectively altering the network's basic-level. This last phase is analogous to a person seeing a pug and saying "That is a pug." This phenomenon of altering the basic-level is observed in adult experts. For example, a layman is more likely to call a pug a dog, while a dog breeder would call it a pug straight away.

Results

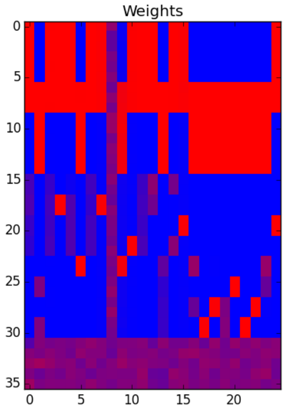


Figure: Execution after 5 epochs (25000 training items total)

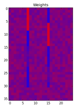


Figure: Execution with IOR disabled after 100 epochs

Finally, we ran an experiment where we completely disabled the IOR operation in an effort to illustrate how crucial it is. As expected, the network was only able to learn basic-level categories for the token objects. Without the IOR operation, the subtle correlations that define the subordinate-level categories are swallowed up by the stronger ones that define the basic-level.

Results

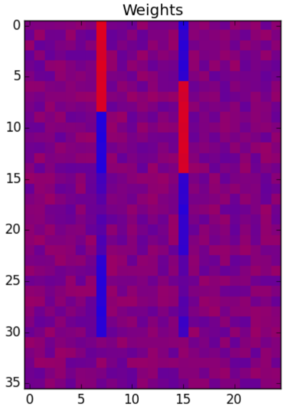


Figure: Execution with IOR disabled after 100 epochs

■ Apply DPAN to real-world image data

1. The next steps for us are to actually apply DPAN to real-world image data. The input we have used so far has just been experimental, abstract toy data. With the proof of concept out of the way, we would like to embed it into a computational vision system to train on actual data.

Future Work

- Apply DPAN to real-world image data

- Apply DPAN to real-world image data
- Investigate how DPAN may handle non-categorical properties (e.g. brown or fluffy)

1. We also would like to investigate how DPAN handles properties, and whether or not they can coexist within the same network. Recall that we brought up the sentences "The dog is a pug" and "The pug". In these sentences the word "pug" can act as either a subject or a predicate, but the same cannot be said for basic properties such as "brown" or "fluffy." We would like to investigate the possibility of encoding these properties as individual units within the same system, so that we would have one unit which represents "pug" and another unit which represents "brown" all in the same network.

Future Work

- Apply DPAN to real-world image data
- Investigate how DPAN may handle non-categorical properties (e.g. brown or fluffy)

■ Each unit has an individual learning rate

If there is time, it is useful to discuss some of components of DPAN that allow it to work so effectively. Each unit has an individual learning rate which decreases based on its gradient of weight change. This prevents units which have "settled" from overlearning. It's a useful metric for measuring how far along training is. If a unit has a level weight change gradient, it basically has learned a category, which means inhibition of return can safely begin.

Additional Components of DPAN's Learning Mechanism

- Each unit has an individual learning rate

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change

If there is time, it is useful to discuss some of components of DPAN that allow it to work so effectively. Each unit has an individual learning rate which decreases based on its gradient of weight change. This prevents units which have "settled" from overlearning. It's a useful metric for measuring how far along training is. If a unit has a level weight change gradient, it basically has learned a category, which means inhibition of return can safely begin.

Additional Components of DPAN's Learning Mechanism

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change
- When the change gradient has leveled off, the unit has reach stability and the learning rate is reduced to zero

If there is time, it is useful to discuss some of components of DPAN that allow it to work so effectively. Each unit has an individual learning rate which decreases based on its gradient of weight change. This prevents units which have "settled" from overlearning. It's a useful metric for measuring how far along training is. If a unit has a level weight change gradient, it basically has learned a category, which means inhibition of return can safely begin.

Additional Components of DPAN's Learning Mechanism

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change
- When the change gradient has leveled off, the unit has reach stability and the learning rate is reduced to zero

If there is time, it is useful to discuss some of components of DPAN that allow it to work so effectively. Each unit has an individual learning rate which decreases based on its gradient of weight change. This prevents units which have "settled" from overlearning. It's a useful metric for measuring how far along training is. If a unit has a level weight change gradient, it basically has learned a category, which means inhibition of return can safely begin.

Additional Components of DPAN's Learning Mechanism

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change
- When the change gradient has leveled off, the unit has reach stability and the learning rate is reduced to zero
- IOR only occurs when the change gradient is level

- Each unit has an individual learning rate
- The learning rates and the IOR starting condition are dependent on each unit's gradient of weight change
- When the change gradient has leveled off, the unit has reach stability and the learning rate is reduced to zero
- IOR only occurs when the change gradient is level

This page intentionally left blank

References

Liu, J., Golinkoff, R. M., & Sak, K. (2001, January). One cow does not an animal make: young children can extend novel words at the superordinate level. *Child development*, *72*(6), 1674–94. Retrieved from `http://www.ncbi.nlm.nih.gov/pubmed/11768139`

O'Reilly, R. C., & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press. Retrieved from `http://books.google.co.nz/books/about/Computational_Explorations_in_Cognitive.html?id=BLf34BFTaIUC&pgis=1`

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976, July). Basic objects in natural categories. *Cognitive Psychology*, *8*(3), 382–439. Retrieved from `http://www.sciencedirect.com/science/article/pii/001002857690013X` doi:

Liu, J., Golinkoff, R. M., & Sak, K. (2001, January). One cow does not an animal make: young children can extend novel words at the superordinate level. *Child development*, *72*(6), 1674–94. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11768139
O'Reilly, R. C., & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press. Retrieved from http://books.google.co.nz/books/about/Computational_Explorations_in_Cognitive.html?id=BLf34BFTaIUC&pgis=1
Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976, July). Basic objects in natural categories. *Cognitive Psychology*, *8*(3), 382–439. Retrieved from http://www.sciencedirect.com/science/article/pii/001002857690013X doi: