HW #1

1. **Why are some of Ken's properties more strongly activated than others?**

   The "Ken" example provides direct(external) input to the Ken name unit. As this unit becomes active, it activates the Ken-in instance unit, which then activates the rest of Ken's properties (H.S., Sharks, Single, Burglar, 20s). These property units, in turn, feed back to partially activate the instance units of other individuals who are similar to Ken in that they share many of his properties. Some of Ken's properties are more strongly activated than others because Nick-in shares (Single, Sharks, H.S.) and Neal-in shares (Single, Sharks, H.S.), they both share 3 mutual properties with Ken so they have a weight of 0.308961 meaning they are fairly active. So some of Ken's properties are more strongly activated than others because the other instance units are partially activated(weaker positive connections/input) compared to being more than partially activated for sharing strong mutual properties(stronger positive connections/input). The activation level of each property unit depends on the amount of net input, so properties like single, sharks, H.S. are more directly connected to Ken-in so they receive positive input from Ken-in's activation and the strong excitatory connections in the weights(+1).

2. **Note that, apart from *Ken-in*, six other instance units—for Nick, Neal, Rick, Earl, Pete and Fred—show differing degrees of partial activation, even though all six individuals share the same number of properties with Ken (three; you should verify this). Why are *Nick-in* and *Neal-in* the most active instance units (other than *Ken-in*), and why is *Rick-in* the next most active?**

   Nick in and Neal in share (H.S., Sharks, Single). Rick in and Earl in share (H.S., Sharks, Burglar) and Pete in and Fred in share (H.S., Single, 20s). Nick-in and Neal-in are the most active instance units after Ken-in because they share Sharks, HS, and Single, they receive strong positive reinforcement from both Ken-in and mutual connections between these properties. Through feedback loops, Instance units with more mutual support from shared properties are activated more strongly because the shared properties reinforce one another. Rick-in is the next most active because it shares Sharks, HS, and Burglar, and Burglar does not connect as strongly with the other properties in the network and faces greater inhibitory competition within the occupation group since it suppresses its activation. The weaker weight of Burglar compared to the other properties limits the overall activation level of Rick-in. Pete-in and Fred-in, sharing 20s, HS, and Single, are less active because 20s lacks strong feedback connections to the rest of the network. The relative activation levels are determine by positive feedback from shared properties, inhibitory competition within property groups, and the strength of mutual reinforcement among properties. Nick and Neal in share Sharks and H.S. with Rick and Earl in and they share H.S. and Single with Pete and Fred in. Rick in and Earl in only share H.S. with Pete in and Fred in and vice versa. Nick and Neal in share more mutual properties amongst the other units as well reinforcing why they are the most active units.

3. **Why do the occupation units show some partial activations of units other than Ken's occupation (Burglar) in the "Sharks 20's" case, whereas they don't in the "Ken" case? [Hint: it will be helpful to examine and compare how activations change over time in the two cases.**

   In the "Sharks 20's" case, the instance units are Pete, Fred, Nick, Neal in and they receive direct input due to shared properties with Ken, where Pete in shares H.S., Single, 20s, Fred in shares H.S., Single, 20s, Nick in shares H.S., Sharks, Single, Neal in shares H.S., Sharks, Single. H.S. and Single is mutual between (Nick in and Neal in) and (Pete in and Fred in), resulting in distributed activation among these units. Nick and Fred in support Pusher, and Pete and Neal in support Bookie, resulting in those occupation units to receive partial activation. After examining how the activations are changing over time in the "Ken" case, Ken-in was the majority basically since it was the only strongly activated instance unit initially until Neal and Nick in later, so the Burglar occupation unit could overtake quickly and suppress activation from the competing units to pass the threshold through inhibition. In "Sharks 20's", temporal dynamics allow more competition and distributed activation, leading to partial activations in other occupation units.

4. **How was the model was able to fill in the correct occupation for Lance? Also, why does the model activate the *Div.* (divorced) unit as well as the *Mar.* (married) unit?**

   The model was able to fill in the correct occupation for Lance because instance units including George in, Al in, Jim in, and John in that share properties with Lance also activate the Burglar unit. So these shared properties create strong positive feedback loops, so it fills in the correct occupation unit for Lance. The model made plausible guesses by similarity through shared properties which supported the activation of associated units. The model activates the divorced unit because George in and Jim in are divorced so it provides partial activation for the divorced unit, since the network uses both bottom up input from properties linked directly to Lance and top down feedback from shared connections resulting in partial activation of Divorced from the overlapping features across the instance units which is an indirect association. The model activated the Married unit because there is a direct association from strong bottom up input to the Married unit and this support is sufficient to overcome the activation threshold. The activation of the Married unit is because Lance's specific features and associated instance units provide direct/indirect support for the marital status. Lance's properties are strongly associated with other individuals in the network and they are also married, so there is bottom up and top down feedback that reinforces the activation of the married unit.

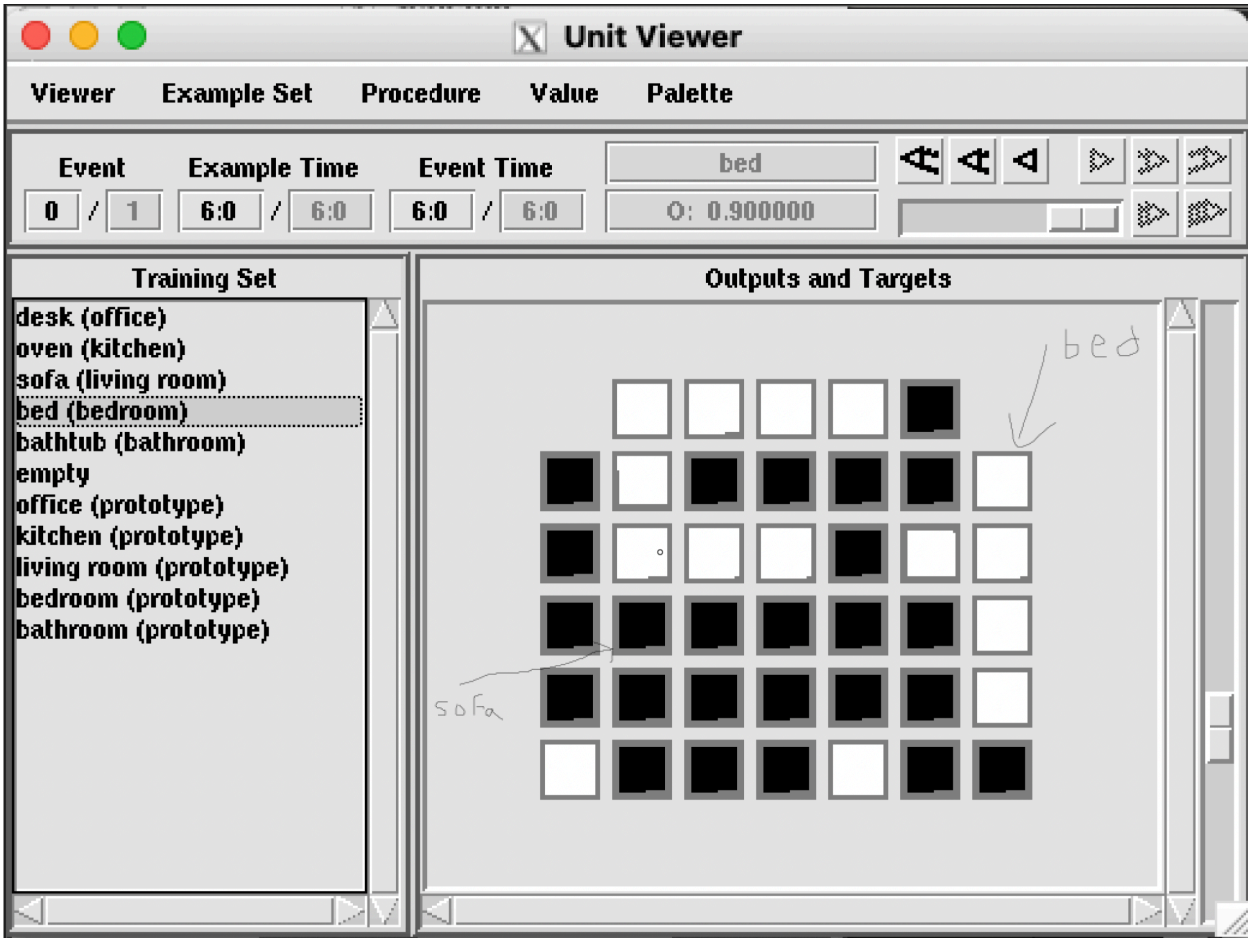
5. **Critically evaluate whether or not the *windows* and *drapes* features do, in fact, form a subschema. Be sure to provide evidence from running the simulation (e.g., goodness values) to support your argument. (Note that Figure 13, in and of itself, is not sufficient to establish their claim.)**

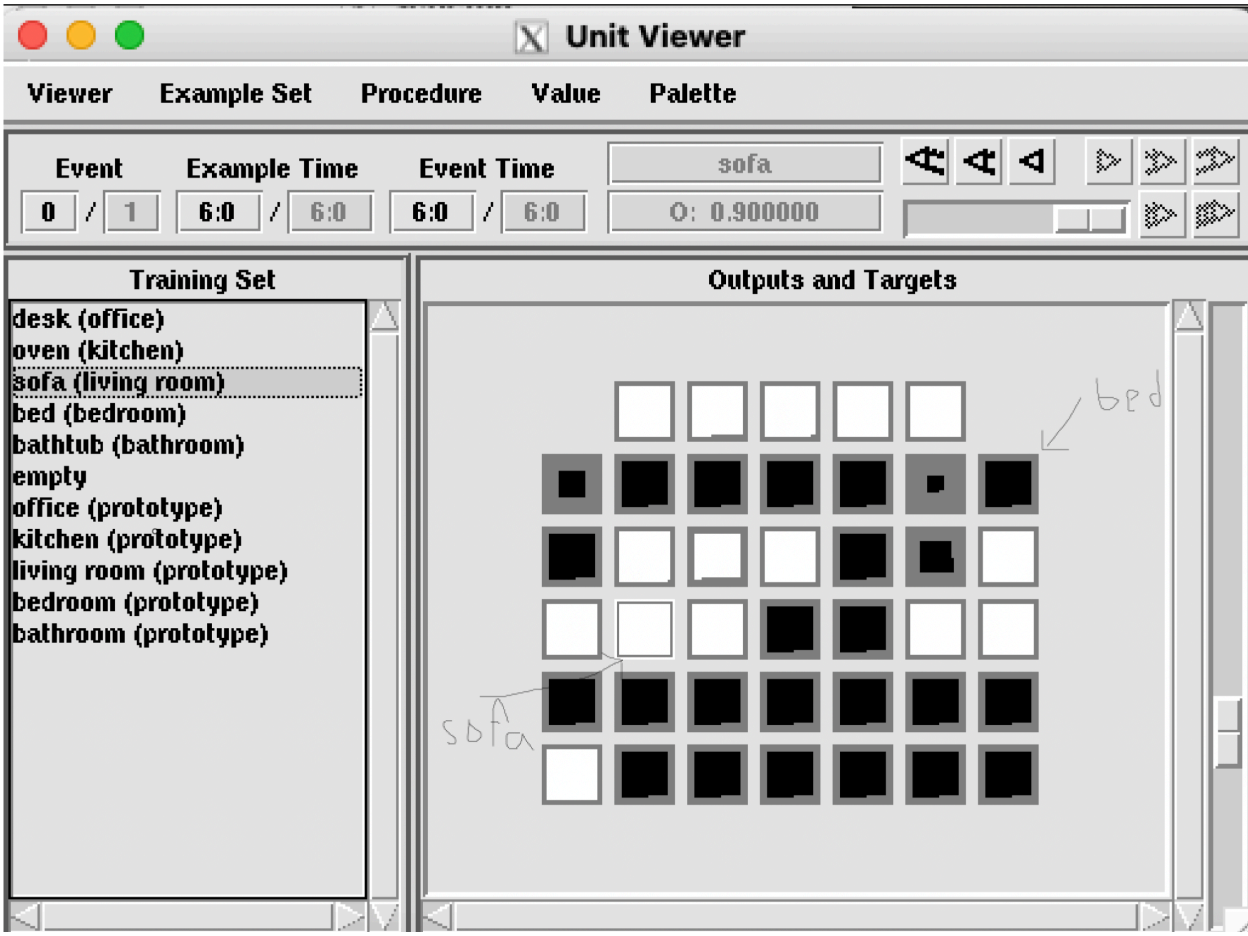
   The windows and drapes features do form a subschema. A subschema is a subset of descriptors that are functionally related and exhibit mutual reinforcement when activated together. When the drapes unit was activated, the highest goodness value was at desk (office) at 20.071752, and the lowest for the bathroom (prototype) at 4.044715. When the window unit was activated it yielded it's highest goodness value of 19.676469 for desk (office) and a lowest value of 3.640116 for the bathroom prototype. When both window and drapes were activated, the highest goodness value for the desk (office) prototype increased significantly to 22.829187, showing mutual reinforcement. This increase in goodness with both activations means that these features interact together and contribute to a mutual interpretation of an office environment. Constraint satisfaction determines subschema relationships so the interaction between window and drapes is from the cooccurrence statistics and shared positive constraints. Bathroom (prototype) didn't increase much because subschemas can embed within larger schemas and maintain their coherence across different contexts, so windows and drapes form a subschema relevant to their specific area. Windows and drapes are more associated with offices than they are in bathrooms. This and the observed patterns of mutual constraint satisfaction, align with Figure 13 where the goodness function is maximized through mutual reinforcement when both window and drapes are either activated together.

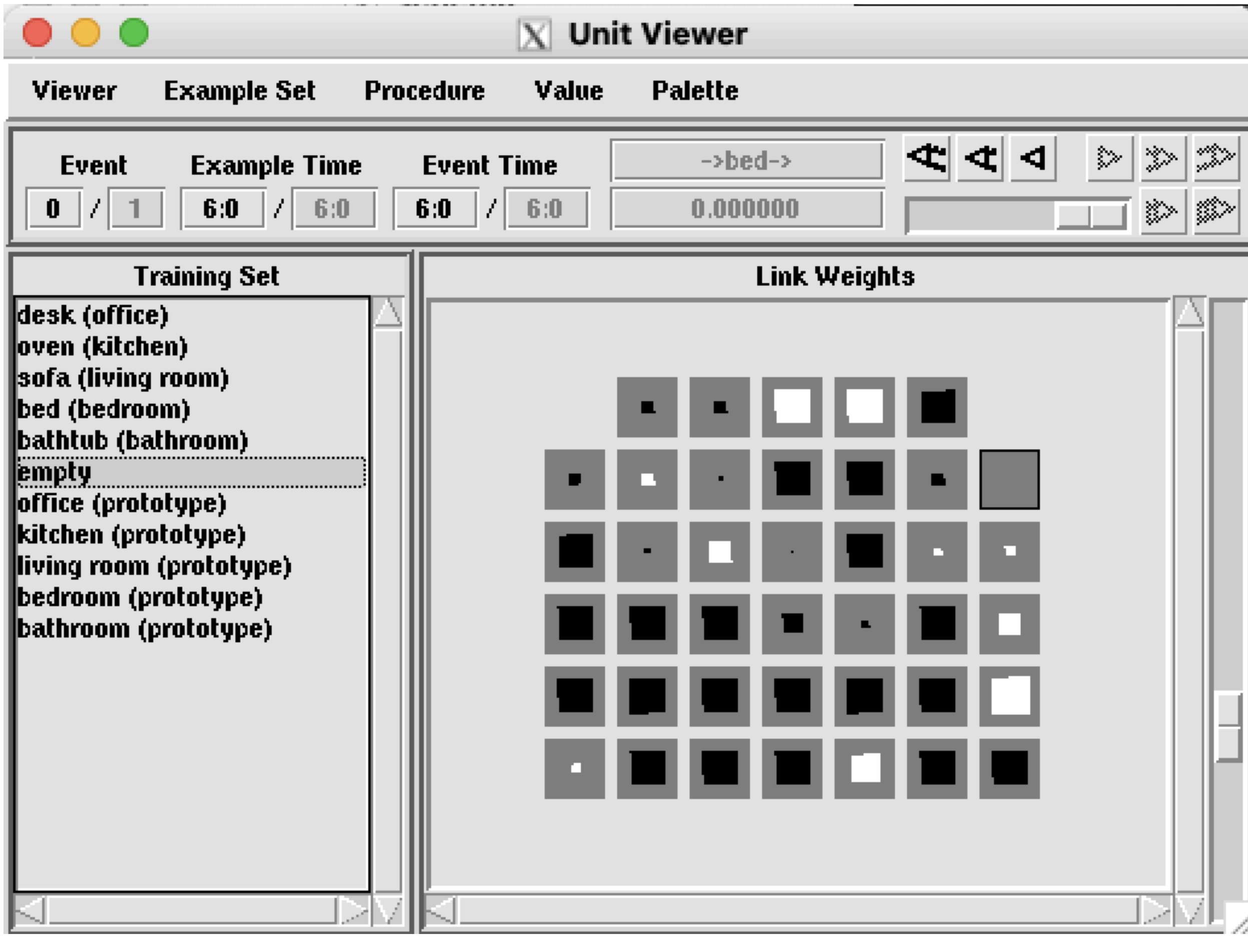
Prototype	Drapes Activated	Window Activated	Both Activated
Desk(office)	20.071752	19.676469	22.829187
Oven(kitchen)	17.019280	16.980356	16.886483

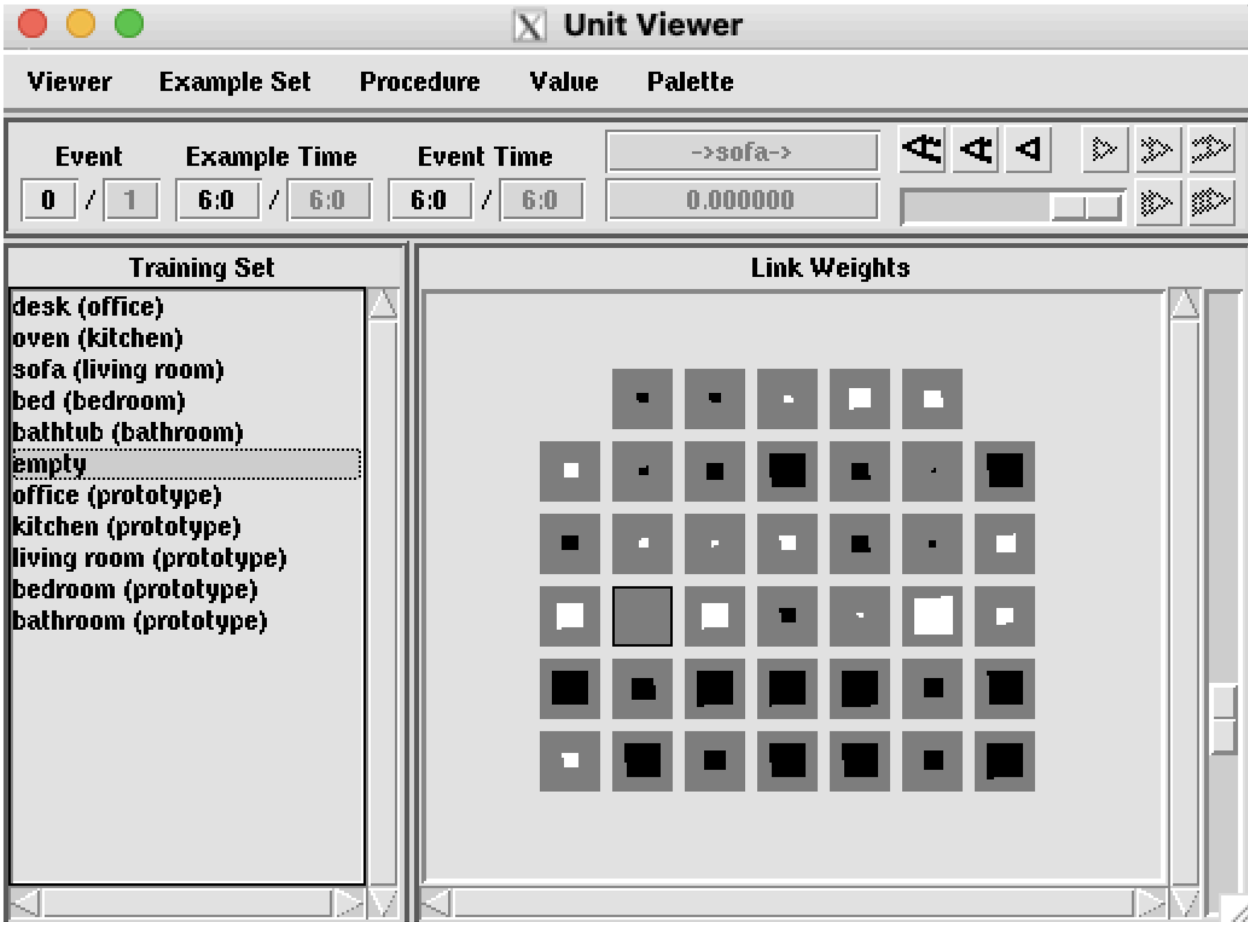
Sofa (living room)	17.325402	17.324617	17.332045
Bed (bedroom)	14.967936	14.967146	14.962791
Bathtub (bathroom)	4.187967	4.166785	4.541666
Empty	17.337204	17.323539	17.341540
Office (prototype)	19.352615	19.109230	19.623804
Kitchen (prototype)	17.203522	17.203522	17.203522
Living Room (prototype)	17.564505	17.564505	17.564505
Bedroom (prototype)	15.050491	15.050491	15.050491
Bathroom (prototype)	4.044715	3.640116	4.541666

6. **Identify the ways in which the combined final pattern differs from each of the single-feature patterns, and try to explain those differences. Does the pattern produced by one of the features predominate, or is the mix fairly even, and why? How does the goodness value of the combined pattern compare with those of the single-feature patterns, and why? You will find it useful to examine the pattern of weights among units (see the actual network or Figure 5 in the Rumelhart et al. chapter) in explaining why the network behaves as it does.**


Unit Viewer
Viewer
Example Set
Procedure
Value
Palette
Event
Example Time
Event Time
0 / 1
6:0 / 6:0
6:0 / 6:0
bed
O: 0.900000
Training Set
desk (office)
oven (kitchen)
sofa (living room)
bed (bedroom)
bathtub (bathroom)
empty
office (prototype)
kitchen (prototype)
living room (prototype)
bedroom (prototype)
bathroom (prototype)
Outputs and Targets
bed
sofa

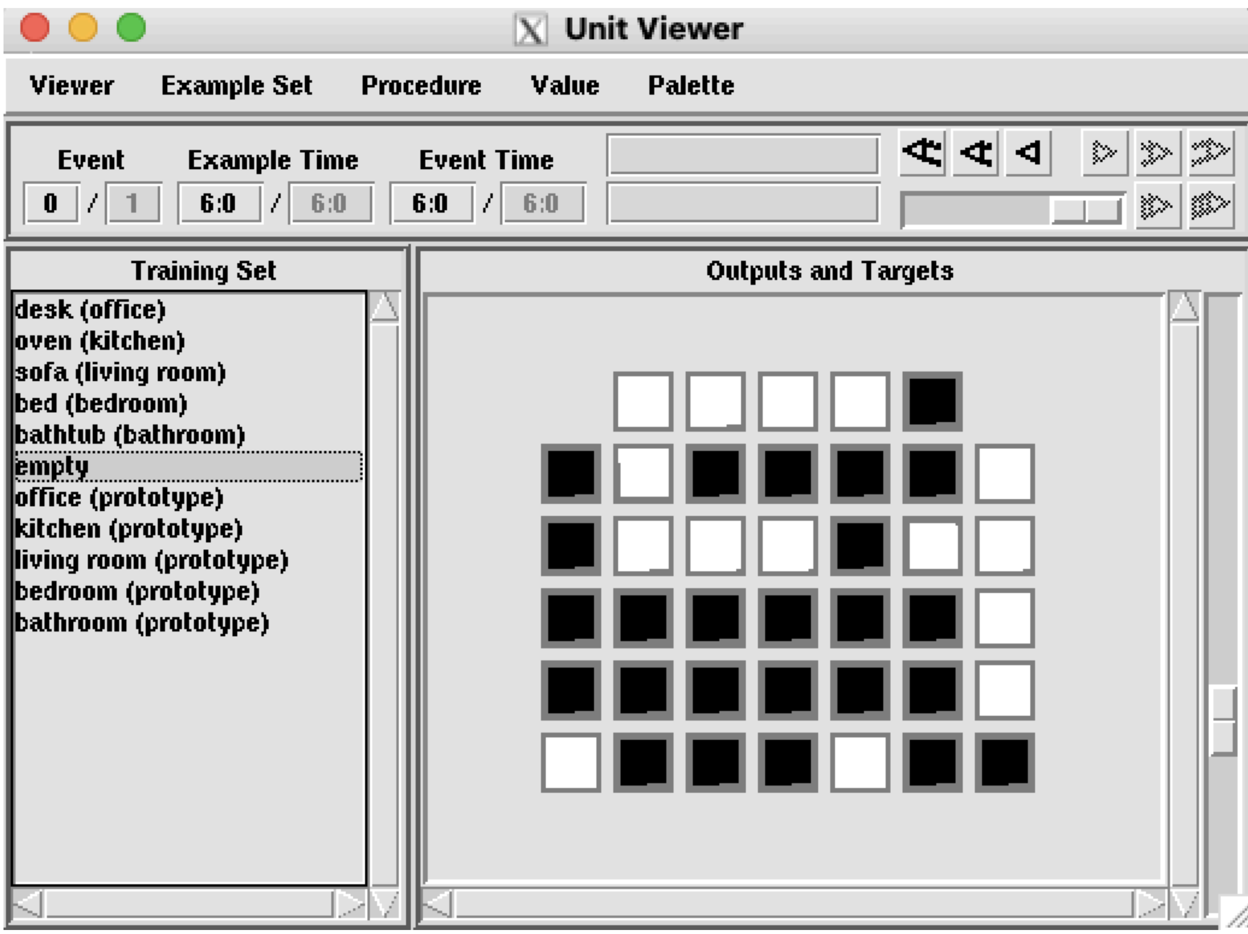

Unit Viewer
Viewer
Example Set
Procedure
Value
Palette
Event
Example Time
Event Time
0 / 1
6:0 / 6:0
6:0 / 6:0
sofa
O: 0.900000
Training Set
desk (office)
oven (kitchen)
sofa (living room)
bed (bedroom)
bathtub (bathroom)
empty
office (prototype)
kitchen (prototype)
living room (prototype)
bedroom (prototype)
bathroom (prototype)
Outputs and Targets
bed
sofa


Unit Viewer
Viewer
Example Set
Procedure
Value
Palette
Event
Example Time
Event Time
->bed->
0 / 1
6:0 / 6:0
6:0 / 6:0
0.000000
Training Set
desk (office)
oven (kitchen)
sofa (living room)
bed (bedroom)
bathtub (bathroom)
empty
office (prototype)
kitchen (prototype)
living room (prototype)
bedroom (prototype)
bathroom (prototype)
Link Weights



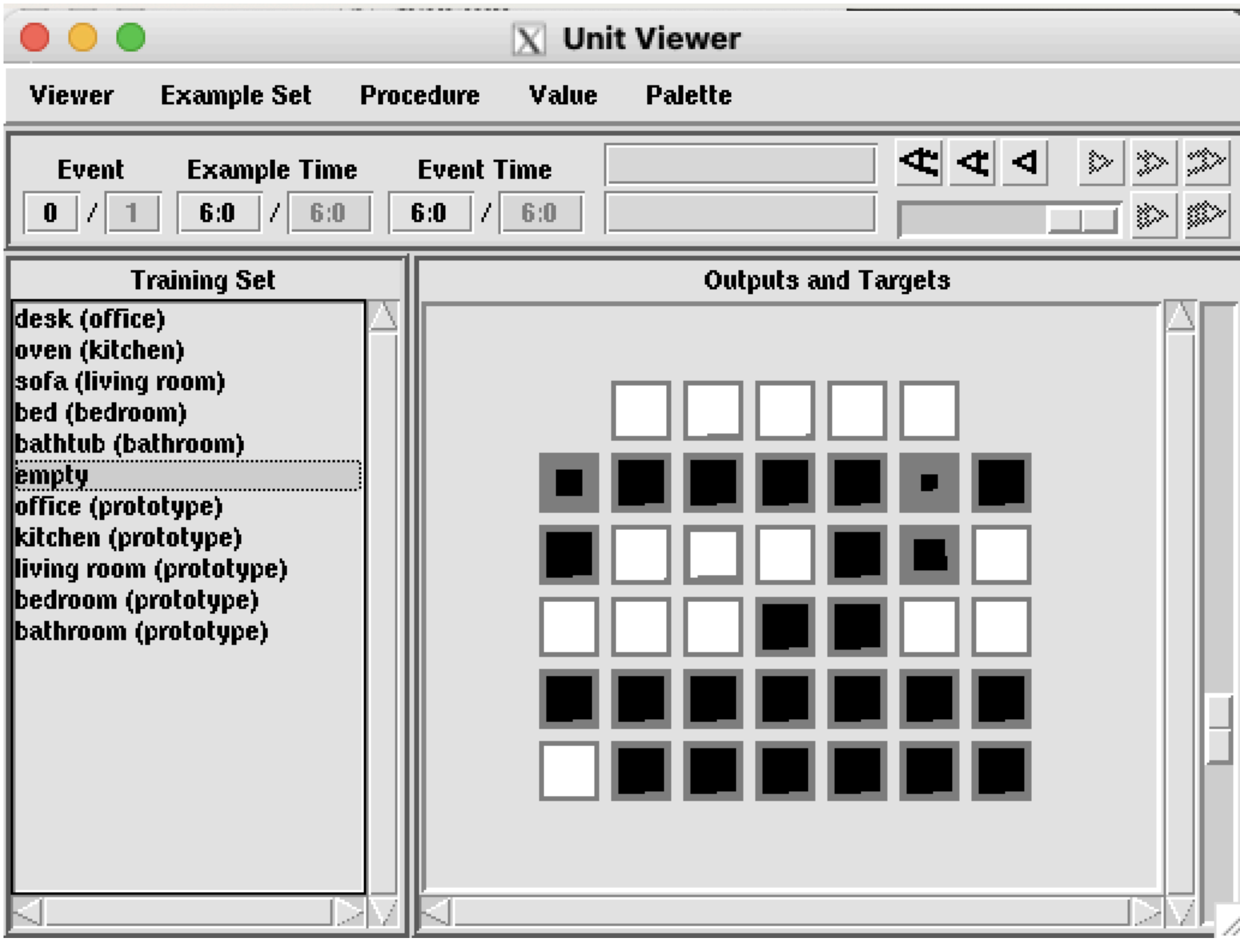
The two features identified which are strongly active in different rooms but are not active together in any of them are bed and sofa. The goodness values and final activation displays are shown below for bed activated, sofa activated, and both activated. When both bed and sofa units are activated it produces a final pattern that is different from bed and sofa individually, but it has some elements of both but doesn't entirely reflect the activation from either individual feature. The combined final pattern has a averaged effect from competition between units that are positively weighted toward one feature and negatively weighted toward the other. This combined pattern has a lower goodness value of 14.993735, when compared to the sofa unit 17.309600, which dominates in strength, and it is slightly higher than the bed unit with 14.948623. The lower combined goodness means that the constraints among features for bed and sofa units are not strongly aligned/associated, resulting in disagreement when both are active simultaneously. The decrease in goodness value is because of the constraints disagreement. The constraint satisfaction mechanism of the network, where the activation of units for bed and sofa simultaneously violates some of the network's learned co-occurrence statistics. The network synthesizes inputs but there are challenges when the inputs represent competing or incompatible subschemas as shown in the Figure 5 description. The pattern produced by the sofa feature predominates in the combined pattern, because of the relative strength of its goodness value and the overlap in features activated in the combined state. This dominance is most likely from stronger positive connections between the sofa unit and its associated features in the network's weight structure, The bed has weaker or competing connections associated with it. The mix is not evenly distributed since the network handles the competition by favoring sofa since it has stronger coherence with the overall goodness landscape. So there are stronger weights associated with the sofa unit, there are more things partially activated overall and in the bed there are mostly inactive units.

Bed activated



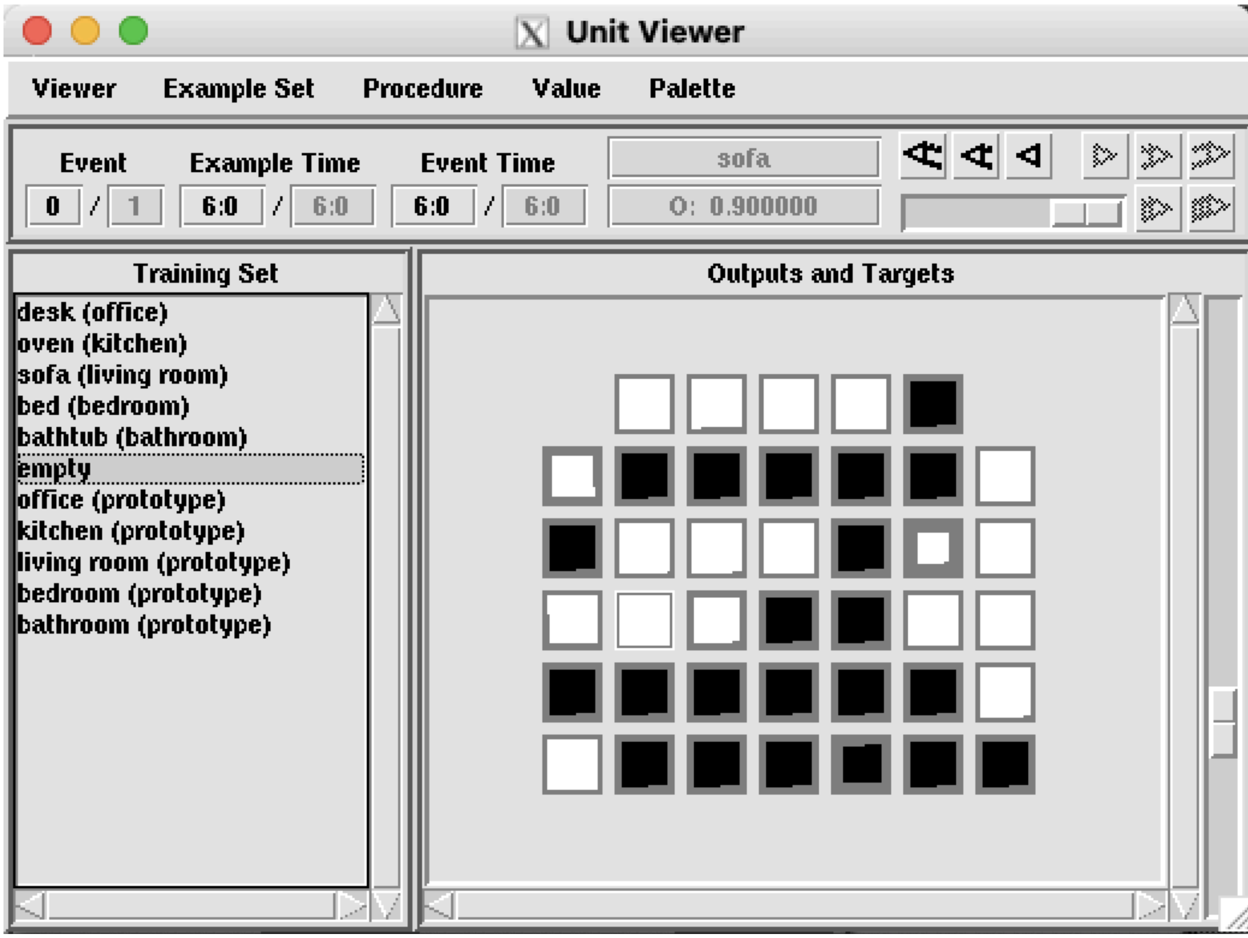
Goodness is 14.948623

Sofa activated



Goodness is 17.309600

Both activated



Goodness is 14.993735