Supervised Human Neural Network

Adam Lenart
alenart@ischool.berkeley.edu
University of California, Berkeley

Matt Shaffer
mattshaffer@berkeley.edu
University of California, Berkeley

December 22, 2017

1 Introduction

Information theory is an useful construct to understand how information moves through a system or network, and is used to model both physical systems and more abstract networks. Social networks are a particularly interesting area of application, but there a number of challenges with quantifying information as it propagates through a human network. When conducting experiments, the self-organizing nature of people into unstructured groups results in complications not limited to spillover effects, unit interference and external biases.

Artificial neural networks have been around for decades, but have recently gained popularity as we enter a paradigm of computing that is defined by what has been called *machine learning.* The basic building blocks of these networks are modeled after individual neurons in the brain, but are configured in structured networks to process complex information efficiently. Although neural networks are still generally not well understood as decision-making mechanisms and considered to be black boxes, the structure under which they operate allows for better error metrics and architectural feedback. These measurements can provide more insight into how information is gained or lost at each network layer and how network design affects coordinated task optimization.

We use this idea of a structured network design from artificial neural networks to impose a deliberate structure on human networks as a way to quantify how strategic arrangement might affect information transmission between units in a coordinated task. We hypothesize that human networks might perform more accurate classifications simply as a result of how units acting as nodes are structurally configured. Following the naming conventions in computer science, we call this network a Supervised Human Neural Network (SHNN), and conduct a synchronized experiment on Amazon's Mechanical Turk platform to evaluate our hypothesis.

1.1 Objective

Here we aim to uncover how information propagates in a structured human network. We structure the network using the analogy of an artificial neural network (Rosenblatt 1958). The goal of the supervised human neural network is to classify food and drug interactions as negative, neutral or positive based on sentences from the biomedical literature.

Our primary outcome is a difference between the accuracy provided by each layer measured by a one-tailed test for equality in two binomial proportions at $\alpha = 0.05$. According to our hypothesis, a higher-order layer has a higher classification accuracy.

2 Experiment Design

2.1 Design Summary

The basic form of the experiment consists of several network layers. First, the participants of the experiment are organized in groups and each participant completes a group-specific classification task. The first layer of classifications that each group provides are used as reference for the subsequent classification tasks. After each participant finishes a task in coordination with a group, the same tasks are shown to the next group of participants along with the classifications from the previous group.

2.2 Details

In order to calculate consensus measures for questions that can be forward propagated to subsequent layers, we devise an experimental design that divides participants into groups that act on data in sequence. These groups are organized into layers, where the first layer passes information to the second, the second the third, and so forth. In order to acquire justifiable baseline measurements and control information spillover between nodes, each group acts on a piece of information exactly once. This means that each question that a group of participants receives is unique in content, and previously unseen.

2.3 Conditions

We also make a distinction between our two experimental conditions. To avoid misuse of the traditional definition of a *control* unit in the literature of field experiments implying a strict counterfactual to the treatment or intervention condition $Y(1)$, we instead define a *baseline condition*, $Y(0)$. This baseline refers to the reference task we ask participants to evaluate, specifically the classification of food and drug interactions from medical literature.

The basic definition of treatment we use can be simplified as having two components: the baseline condition (described above) paired with additional network information. After each layer interacts with a classification example, the classifications are aggregated as a mean estimate and presented to the next layer along with individual estimates. The graphical presentation of this information can be found in the appendix.

The number of groups (G) and unique questions (Q) are determined by the number of network layers (L) we wish to measure, informed by our power calculations. G and Q are a function of $L + 1$, where 1 is the number of the baseline layers before treatment begins.
.

2.4 Baseline Layer

Since the first layer of the network can functionally have no previous estimates, it serves as the baseline measurement. Under no-anticipation assumptions, this first layer of the network also has no prior knowledge of the treatment to be received, which adds to the validity of the estimates as a point of reference that serves analogously to a control condition[1]. Participants work independently in this layer and should provide unbiased estimates that are free of interference from other participants.

2.5 Treatment Layers

With one baseline layer at l_0 we use each layer following in the network as a treatment layer where each classification question is given to a group with the previous estimate. As a question forward propagates to deeper layers in the network, it gets filtered by more groups on its way to the final classification. This creates a nuanced definition of the treatment condition in that the estimates at each layer are theoretically different to layers at time steps l_{t-1} and l_{t+1} if information gain or loss is present.

2.6 Power calculations

Setting up power calculations for comparing the accuracy of network layers is not straightforward, and we had to rely on Monte Carlo experiments to achieve them.

In the power calculations, each simulated trial consisted of several layers with the first layer being the reference layer in which each participant, organized in groups, would provide classification for a set of sentences. After the participants have finished their classifications, they would move to the next layer where they are asked to classify another set of sentences which was already classified by another group of participants in the reference layer. As treatment, they were shown information on the classifications in the preceding layer. Following this method, we stacked layer upon layer.

Let $\mathfrak{S}$ be a binomially distributed variable denoting the number of successful classifications as

$$\mathfrak{S} \sim Bin(n, p)\,,$$

where n stands for the number of classification tasks and p the probability of a successful classification, respectively. Furthermore, as p is a realization of the participants aptitude for a correct answer, and is likely to differ among participants or even within participants for different questions, let p be a random variable as well as

$$p \sim Beta(\theta\pi, \theta(1-\pi))$$

with θ denoting overdispersion and prior probability π. A beta distributed prior for a binomial probability yields a beta-binomial conjugate posterior distribution for S successes (see example Figure 3):

$$S \sim Beta - Binomial(n, \theta\pi, \theta(1-\pi))\,.$$

[1] As we acknowledge in the results, while this should have been true, at runtime MTurk workers were not prevented from participating only once and our samples include repeat workers.

Let s be a realization of S, then the layers update their prior probability in a Bayesian manner by $\pi := \frac{s}{n} + \psi$ where ψ creates a drift in the random process. It can also be understood as a parameter for optimism that humans would utilize network information better than expected from beta-binomial random draws.

Then layers can be compared to each other by a two sample test for the equality of binomial proportions. The power calculations show that primarily the Based on the power calculations depicted in Figure 2, we expect to be able to identify significant differences with groups of at least 20 participants with a power of 60%s assuming a drift (optimism) of 5 percentage points improvement in classification accuracy by layer, or 90% assuming a drift of 10 percentage points improvement by layer.

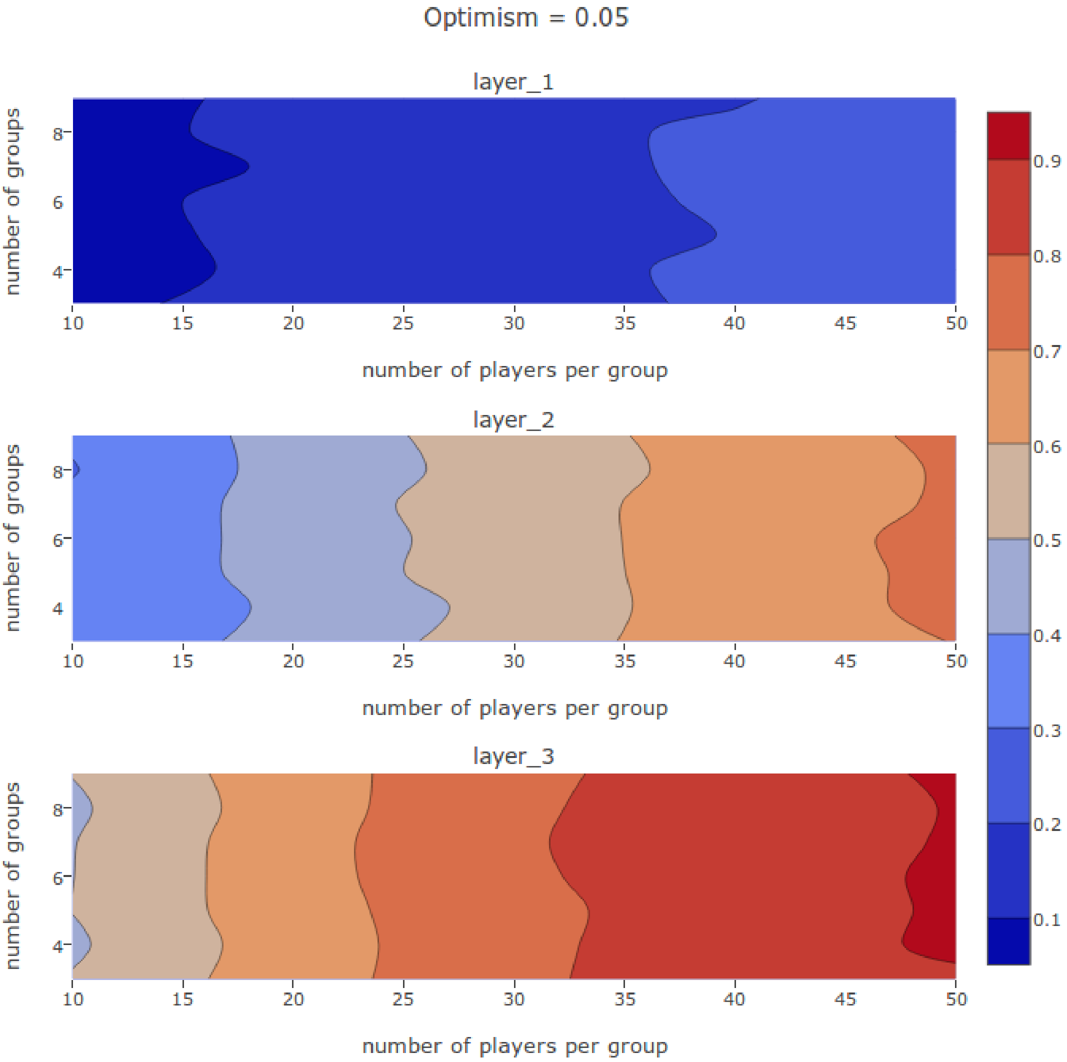


Figure 1: **Power calculations** Each panel shows the power of a one-tailed two sample binomial proportion test over number of players per group and number of groups for each layer using an optimism parameter of 0.05 as drift.

2.7 Classification task

The experiments task was to classify sentences describing food and drug relationships into negative, neutral or positive classes. These sentences were coming from over 300,000 abstracts downloaded from PubMed corresponding to 12 drug classes (Lenart et al. 2017). The abstracts were further filtered by selecting only sentences where a drug class co-occurred with a food compound found in FooDB.ca. Finally, in total, they classified 2,467 sentences containing both drug classes and food items into three classes: *positive*, *neutral* or *negative*.

Out of these 2,467 sentences, we selected sentences that were 100-250 characters long, and as the neutral class was over-represented in the sample, we downsampled it to arrive at a final sample of 65 positive, 86 negative and 72 neutral sentences. We used the character length of a sentence as a crude proxy for the difficulty of the classification task. See Table 4 in the appendix for examples.

2.8 Participants

The participants for creating a human neural network were recruited on an online labor market, Amazon Mechanical Turk. While the quality of responses coming from Amazon Mechanical Turk workers, or turkers, can be disputed (Goodman et al. 2013), there has been considerable interest in experiments on Amazon Mechanical Turk in the social and behavioral science literature (e.g., Crump et al. 2013; Goodman et al. 2013). Paolacci et al. (2010:416) claim that "Mechanical Turk is a reliable source of data in judgment and decision-making. Results obtained in Mechanical Turk did not substantially differ from results obtained in subject pool at a large Midwestern U.S. university. Moreover, response error was significantly lower in Mechanical Turk than in Internet discussion boards." Beside running experiments on the turkers, Mechanical Turk is often used for tasks such as classifying images or sentences for supervised learning algorithms. (Callison-Burch and Dredze 2010; Rashtchian et al. 2010).

2.9 Randomization

To escape the correlation of treatment assignment with the outcome variable, we create independence between units by randomizing both questions and participants. The basic randomization scheme is described as follows, but required some modifications based on the experimental platform that we elaborate on in the section about Mechanical Turk.

Upon their arrival to the experiment, participants enter a virtual waiting room. Under ideal conditions [2], the rooms fills to the capacity needed to reach the number of participants equal to G. We then randomize all participants into groups and randomly select a baseline question for each group.

After the baseline layer, the questions are shifted by an index of 1 in the group assignments, so that if the second group saw the question($id = 314$), the third group would see this question in the first treatment layer. Following this strategy, the fourth group would see question($id = 314$) in the second treatment layer and so on. Since both questions and groups have been randomly assigned, the layer assignment is also random by recursion, even though we systematically move the questions between groups at each layer.

[2] Ideal conditions being those where participants are not more likely to attrit due to longer waiting times.

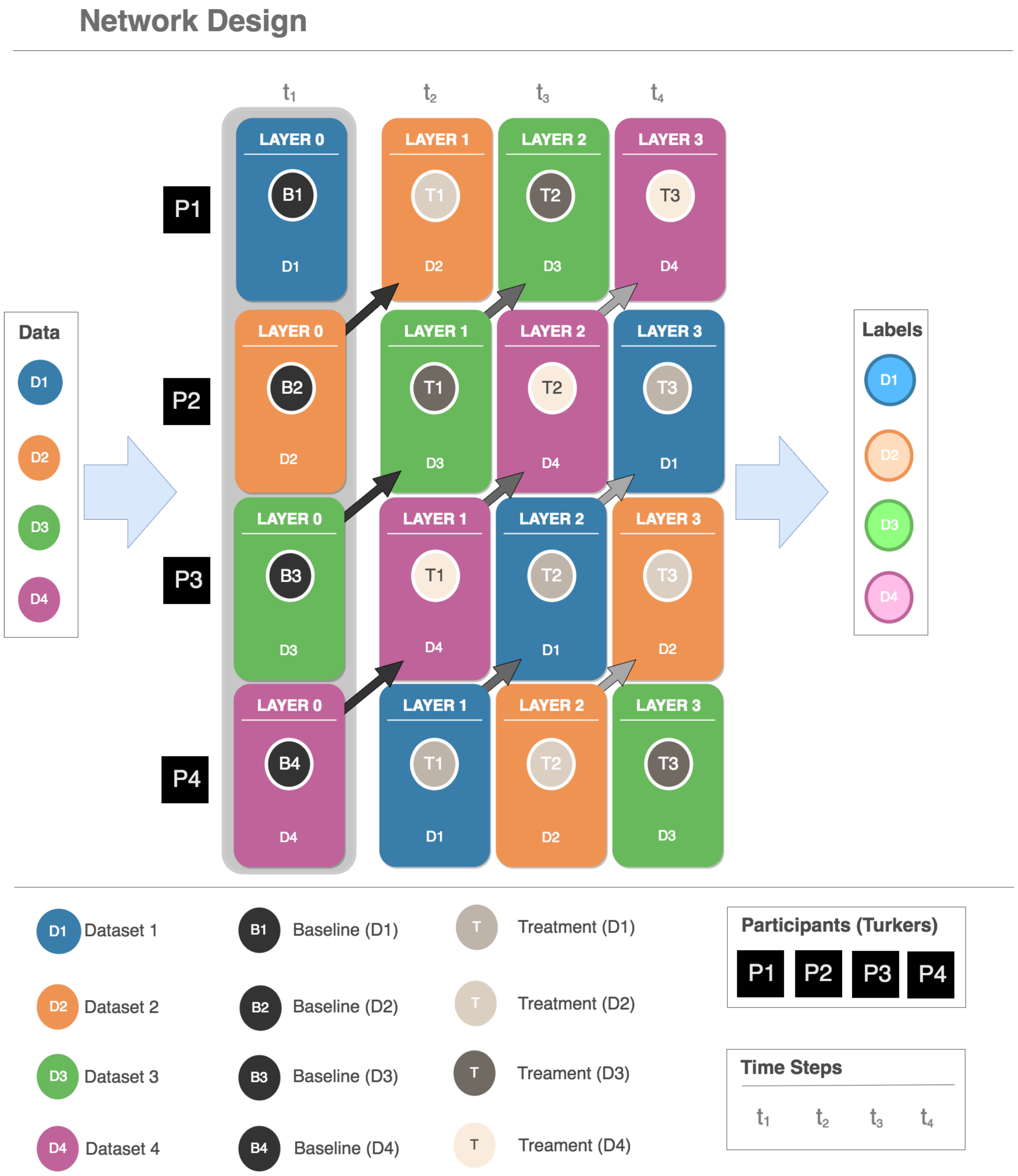


Figure 2: **Network Design and Randomization Procedure** The diagram shows how tasks are assigned to each layer in a four layer network where the first layer serves as the baseline. Each time step (t_i) represents a network layer. Both question data (represented as $D_1, ..., D_4$) and participants ($P_1, ..., P_4$) are randomized prior to the baseline layer. Note that only four participants are represented, but in implementation each participant $P_1, ..., P_4$ belongs to a different group $G_1, ..., G_4$ comprised of approximately 20 participants.

3 Treatment Delivery

3.1 User Interface Design

The effectiveness of delivering network information as a treatment is dependent on many factors including the user interface, and may affect whether or an effect can be detected when measuring the outcome. For instance, considering that existing work on information theory relies on quantifying information via estimates of entropy, we can imply that all information representations are not equal. A picture is different than a thousand words not only in form, but concept and context.

We provide three types of information representations in both baseline and treatment conditions: text, quantitative, and graphical. This allows the participant to choose the representation that is most useful from their perspective. A description of each element follows:

1. Text
   - Previous estimates are qualified as being ”Negative”, ”Neutral” or ”Positive”
2. Quantitative
   - Participants are instructed to give their estimate with a score that corresponds to the confidence of a given classification in a range [-100, 100]
3. Graphical
   - The corresponding scores are accompanied by a color intensity representation where Blue is ”Negative”, White is ”Neutral” and Red is ”Positive”. Estimates that fall in a range between extremes are gradients.

Using these design elements to construct an user interface, we give participants a way to represent their classification estimate as well as their uncertainty. Classifications are performed using a slider element with a continuous value range and indications of class designation based on position values on the x-axis (see Figure 4 in the Appendix for examples). The rationale here is that we provide a way to overcome the absence of back propagation by using network certainty as an error-correcting mechanism. It reasons that more certainty within the network (e.g. higher confidence) would exert more persuasion on an individual node to conform and provide a similar classification. Over time, we would expect this estimate to move closer to an equilibrium estimate that represents a wide range of perspectives.

In addition to allowing participants to represent levels of uncertainty, the slider element can also be programmed to pass information to the next layer by setting its default to the mean estimate from the previous layer. This leverages popular ideas around behavioral psychology that involve ”nudging” by suggestive defaults(Samuelson and Zeckhauser 1988), whereby people tend to be persuaded toward an action that arrives in a default state, or ”anchoring” which gives them a reference point by which to make a decision(Tversky and Kahneman 1992).

While these user interface design decisions require assumptions about efficacy, we chose the ultimate design based on our understanding of how each element might deliver the most effective treatment.

4 Mechanical Turk Considerations

Experiments conducted on Amazon's Mechanical Turk platform have many systematic challenges, and this section describes some of the considerations made when using it to conduct the experiment. Real-world field experiments are prone to bias from a variety of externalities that can occur, and although lab experiments can be more resistant to the effects of uncontrolled influences, Mechanical Turk is not explicitly a laboratory environment.

4.1 Attrition

One characteristic of Mechanical Turk experiments is that they are prone to high rates of attrition. Workers can chose to move from one assignment to another and often dropout, especially if there is waiting involved. Conducting a large scale synchronous experiment on the platform is therefore not ideal, and managing attrition required adapting incentive structures, implementing timing mechanisms and doing compliance checks before randomizing.

Taking the prevalence of attrition into account, the software platform used to conduct the experiment (Chen et al. 2016) defaults to recruiting twice the amount of experimental subjects as defined in the settings to ensure that experiments are not underpowered. While this strategy may work well for small scale economic experiments with fewer participants, this led to a wide range of uncertainty when estimating compliance after HIT acceptance and payouts based on unpredictable dropout rates.

4.2 Timing and Timing Mechanisms

Waiting times require strategic management on MTurk, and there are several details of our implementation that served to manage waiting times and set expectations for workers. The most important of these was to set the round progression on a global timer and schedule the experiment start time precisely in Universal Coordinated Time (UTC). Both the title and description of the HIT specified this start time, and a countdown timer indicated the exact wait time remaining. This meant that Turkers would not be waiting for an experiment queue to fill for an undetermined length of time, a common problem that was reported to increase attrition in other experiments.

Each round was also subject to the global timer, which kept the experiment on schedule and gave participants a visual indication of countdown to when the next round would begin if they finished a question early. The time to complete each question was set to 90 seconds following pilot testing. One downside to this method is that the actual time a Turker has to make a classification estimate may vary based on client-side loading times or connectivity differences. We expect that the randomization procedure would minimize any unbalancing effect between groups, but there is a chance that differences might exist due to a small sample size. So, since the baseline measurements for a given question are measured for a different group than receives treatment for that same question, we might observe group differences in mean page loading time that could affect classification accuracy.

Another complication that results from the global timing strategy is that there is a chance that the experiment may begin before an adequate number of participants enter the experiment queue. Worker demand for an assignment is contingent on many external factors such as time of day, day of week, HIT description, requester reputation, and incentives offered – all of which are related to a variable intensity of competition from other MTurk requesters. This makes predicting the available pool of workers difficult, and while the easiest way of overcoming variability of population availability is to offer higher incentive payout, this is not practical on an academic research budget. As a result of this unpredictability, our final run of the experiment was conducted with a smaller number of participants than expected, and limited the amount of network layers we could construct based on the predetermined group size parameter we defined.

4.3 Incentives

Turker expectations for HIT compensation generally fall in the range of about \$6-\$12 an hour although actual compensation is generally lower (Stewart et al. 2015). Based on these benchmarks, we estimated we would need to offer a competitive rate on the higher end of the scale to overcome the anticipated attrition. We initially set incentives at \$1.75 for a completed HIT, with a bonus per question answered correctly of \$0.25. With a six layer network (including the baseline layer) this resulted in a pay range of \$1.75 for zero correct answers to \$3.25 for 6 correct answers with a minimum expected payout of \$2.25 per worker based on random guessing. The entirety of the HIT would take 10-15 minutes.

The philosophy of offering a bonus-based reward system served multiple purposes. First, it was thought to motivate workers to tolerate the wait times in the experiment and encourage accuracy over completion speed, which is a naturally-occurring intrinsic motivation provided by the MTurk platform. Since Turkers are compensated on a per HIT schedule, they might have been motivated only to choose estimates at random unless it was economically beneficial to get correct answers. Second, in treatment rounds where collaborative behavior is desired, they might be more trusting of network information if they believed all workers were working toward a common goal. This would conceptually equate to information that was more valuable and analogous to administering a higher dosage of the treatment. Third, payout expectations could span a range that might be more attractive to workers in a competitive market for workers.

A first attempted run of the experiment confirmed this as a competitive rate, and all available slots were filled in the experimental queue within minutes of posting. Unfortunately, technical problems prevented the experiment from executing properly. In following runs, the base rate was lowered to \$1.25, but the HIT acceptance rate was much lower. The timing of the experiment may have been a factor though, since the first attempt was made on a weekend when reports indicate there may be fewer HITs available to workers and less competition from other requesters.

4.4 Active vs. Inactive Workers

As a result of differential attrition rates on MTurk, we make a distinction between *active* and *inactive* participants to ensure that information is propagated effectively from layer to layer. The way HITs are conducted on the platform means that Turkers are assigned a unique session id as soon as they choose to accept the HIT for the experiment and enter the virtual waiting area which shows the countdown timer described earlier. This is the

layer	# of correct	n	accuracy	p
reference	48	123	0.39	-
treatment.1	46	115	0.40	0.49
treatment.2	24	52	0.46	0.24

Table 1: **Accuracy by layers**. The p-value corresponds to the null hypothesis that the proportion of correct answers in a treatment layer if greater than the proportion of correct in answers in the reference layer.

most likely time for attrition to occur due to the length of time that waiting is required, but also because the instructions for the task are detailed here and there may be a degree of self-selectivity wherein workers decide whether the task is interesting to them or not. For example, if a worker sees that the classification task appears too difficult, it may affect their expected incentive payout and cause them to leave the queue. Unfortunately, if this happens, there is no way to reassign the session id, and the slot remains vacant for the duration of the experiment.

We use this interim page between HIT acceptance and experiment start time as a way to sort users as active or inactive based on their response to an *acceptance of terms* radio selection at the bottom of the page. Compliance of this additional step means that we classify a worker as ”active” and allow them to participate in the experiment. As pages are programmatically advanced, this acceptance also helps prevent workers from accepting the HIT and leaving their browser open while the experiment moves toward completion automatically allowing them to collect payment for a single mouse click.

At the start time of the experiment, active/inactive assignments are finalized and the randomization procedure takes place as previously detailed with one exception based on acceptance of terms criteria: only active players are randomized into groups that will become the layers of the SHNN. Since the total amount of active participants is unlikely to be exactly divisible by the group number hyperparameter, the remaining active participants that do not get assigned to a group are combined with inactive players and assigned only randomized questions under the baseline condition at all layers. Inactive players were assigned randomized questions in early runs, and completely blocked from participating in the final run as a convenience to aid in tallying Turker payout later. [3]

5 Results

According to the objectives and the subsequent power calculations, we aimed at uncovering significant differences between baseline and treatment layers. We have performed two experiments with 87 and 53 participants, 19 participants took part in both experiments[4]. As the participants answered the questions on a $[-100, 100]$ scale, we fit a k-means ($k = 3$) clustering algorithm for identifying the labels and, as the answers were relatively uniformly distributed on the scale with a spike at 0, the clustering algorithm identified negative labels to be between $[-100, -29]$, neutral $[-28, 33]$ and positive $[34, 100]$, respectively.

As Table 1 shows, the binomial proportion test fails to reject the null hypothesis that the

[3] In later runs, we also omitted one complete active group from the treatment layers for the duration of the experiment and assigned only randomized control questions as an additional baseline measure.

[4] Interestingly, they had a somewhat lower than average classification accuracy)

accuracy increases from the reference to a treatment layer. However, this analysis assumes that the questions that the participants received are of the same difficulty. To test the assumption that the different labelled questions might have a different difficulty level, we added a logistic and Poisson regression for estimating the odds of a correct classification and the number of correct classifications while offsetting for the natural logarithm of the number of answers, respectively. In Table 2, both models indicate that the positive questions may have been easier to answer than neutral or negative ones. Focusing on the effect sizes, they also indicate that the class of the question might be a more important predictor of success than the treatment layers.

	Poisson	Logistic
(Intercept)	-1.20^{***}	-0.85^{**}
	(0.21)	(0.26)
treatment.1	0.03	0.06
	(0.21)	(0.27)
treatment.2	0.06	0.12
	(0.25)	(0.34)
neutral	0.30	0.46
	(0.23)	(0.29)
positive	0.57^{*}	0.99^{**}
	(0.25)	(0.34)
Num. obs.	9	290

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2: **Checking classification difficulty by label class** Poisson and logistic regressions for predicting correct class labels.

If we look at the data in more granularity, the difficulty of the questions show a large variance (Figure 4) as some questions seem to be harder to answer than others. However, accounting for question-level fixed effects still fail to reject the null hypothesis that the allowing turkers to see the decision of other turkers increases the classification accuracy (Table 3).

	Poisson
(Intercept)	-0.74^{**}
	(0.25)
treatment.1	-0.12
	(0.26)
treatment.2	0.19
	(0.28)
including question-level	
	fixed effects
Num. obs.	17

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 3: **Classification accuracy on the question-level**. Poisson regression with question-level dummy variables and the natural logarithm of number of answers as offset.

6 Conclusion

The potential to explore how network configurations affect information processing by human "nodes" raises interesting questions about how humans share, store, and contribute to decision-making accuracy in cooperative environments. We believe the conceptual basis for this experiment has a theoretical grounding warranting further exploration but have demonstrated the difficulty of implementing such a coordinated experiment on a platform like Mechanical Turk. After several attempted runs, the complexity of the experimental design must be reconsidered, or the technology revised. Based on the sample sizes of the experiments that were conducted, we could not expect to find statistically significant effects but in general the coefficient estimates point in the direction of the hypothesized effect.

7 Remarks

We urge any lecturer at a higher educational institution, who has access to a large number of students that can be prevented from attrition, reading this to try the experiment with the students. The code implementation is publicly available on Github(Lenart and Shaffer 2017), and undergraduates present in the same location might be a more suitable environment for interactive network experiments than Amazon's Mechanical Turk platform.

References

Callison-Burch, C. and M. Dredze. 2010. "Creating speech and language data with Amazon's Mechanical Turk." In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 1–12, Association for Computational Linguistics.

Chen, D. L., M. Schonger and C. Wickens. 2016. "oTree—An open-source platform for laboratory, online, and field experiments." *Journal of Behavioral and Experimental Finance* 9:88–97.

Crump, M. J., J. V. McDonnell and T. M. Gureckis. 2013. "Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research." *PloS one* 8(3):e57410.

Goodman, J. K., C. E. Cryder and A. Cheema. 2013. "Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples." *Journal of Behavioral Decision Making* 26(3):213–224.

Lenart and Shaffer. 2017. "Supervised Human Neural Network." https://github.com/planetceres/SHNN.

Lenart, A., H. Moon and L. Barceló. 2017. "Evaluating Food-Drug Interactions with Artificial Neural Networks." Unpublished manuscript.

Paolacci, G., J. Chandler and P. G. Ipeirotis. 2010. "Running experiments on Amazon Mechanical Turk." *Judgement and Decision Making* 5(5).

Rashtchian, C., P. Young, M. Hodosh and J. Hockenmaier. 2010. "Collecting image annotations using Amazon's Mechanical Turk." In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147, Association for Computational Linguistics.

Rosenblatt, F. 1958. "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review* 65(6):386.

Samuelson, W. and R. Zeckhauser. 1988. "Status quo bias in decision making." *Journal of Risk and Uncertainty* 1(1):7–59.

Stewart, N., C. Ungemach, A. J. L. Harris, D. M. Bartels, B. R. Newell, G. Paolacci and J. Chandler. 2015. *The Average Laboratory Samples a Population of 7,300 Amazon Mechanical Turk Workers*. Mathematica Policy Research Reports, Mathematica Policy Research, URL https://EconPapers.repec.org/RePEc:mpr:mprres:f97b669c7b3e4c2ab95c9f8051d18af6.

Tversky, A. and D. Kahneman. 1992. "Advances in prospect theory: Cumulative representation of uncertainty." *Journal of Risk and Uncertainty* 5(4):297–323.

A Appendix

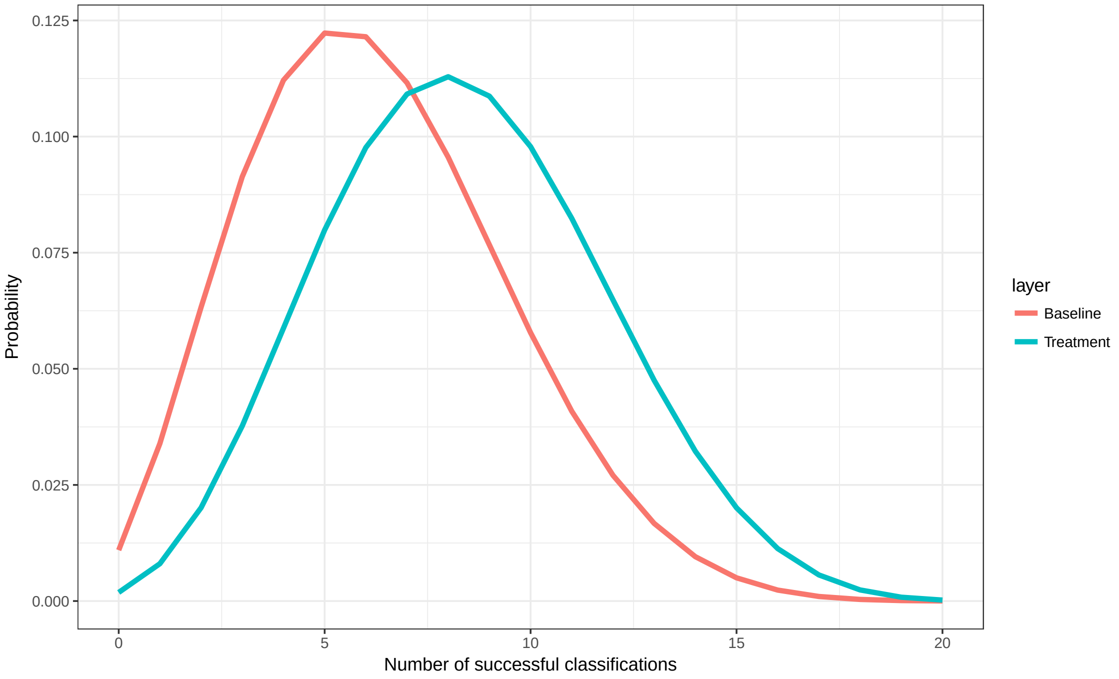


Figure 3: **Power calculations example** Two example beta-binomial distributions showing the probability of number of successful classifications in baseline (red) and treatment (blue) layers.

Interaction	Food item	Drug compound	Sentence
Negative	unsaturated fatty acids	isoniazid	Both long- and short-term exposure experiments showed that isoniazid inhibited the synthesis of of saturated fatty acids greater than C26 and of unsaturated fatty acids greater than C24.
Neutral	vitamin D3	digoxin	The median for digoxin T(max) was 0.75 h before and after vitamin D3 ingestion.
Positive	curcumin	GLP-1	Our previous studies demonstrated that curcumin (a yellow pigment of turmeric) significantly increases the secretion of GLP-1 in enteroendocrine L cell line (GLUTag cells).

Table 4: **Classification tasks** Example sentences for classifying the relationship between a food item and a drug compound

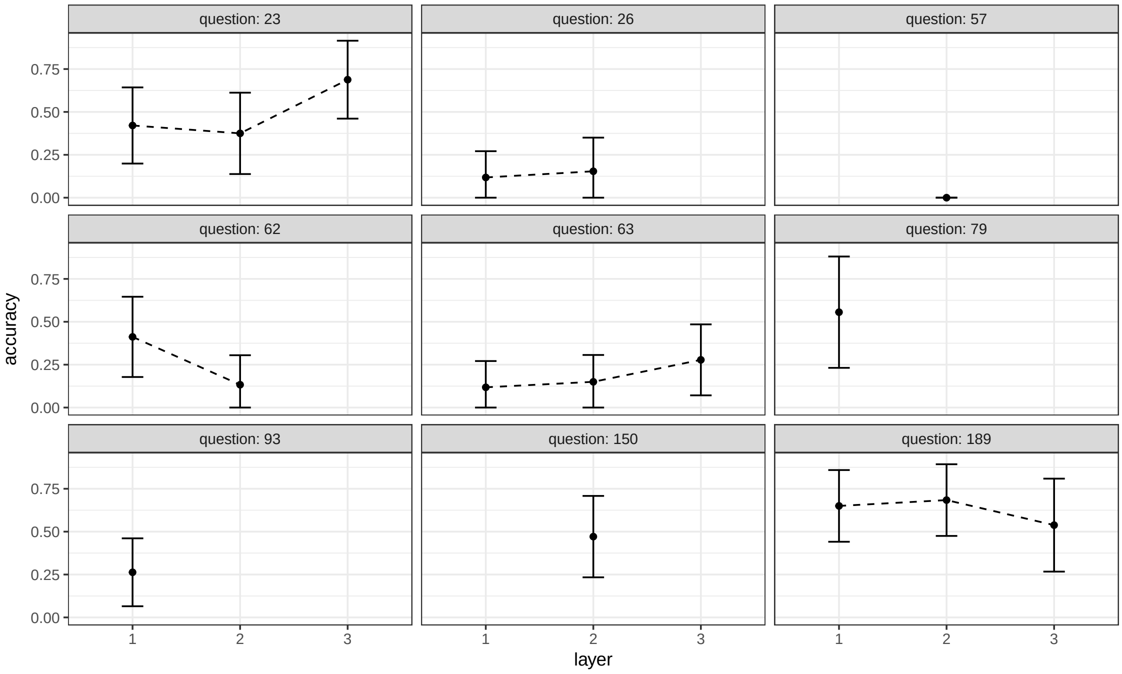
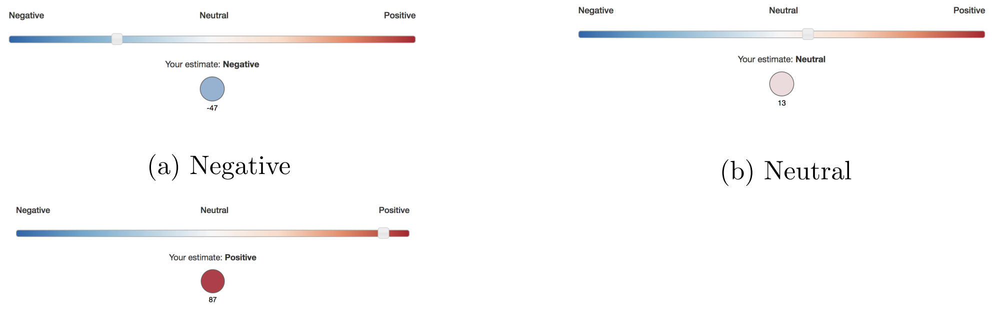


Figure 4: **Classification accuracy by question and layer** The first experiment involving 87 participants could include reference plus two treatment layers, however, the second, smaller experiment with 53 participants could only accommodate one baseline and one treatment layer. Several questions were answered by only one layer due to problems with attrition during the experiment.

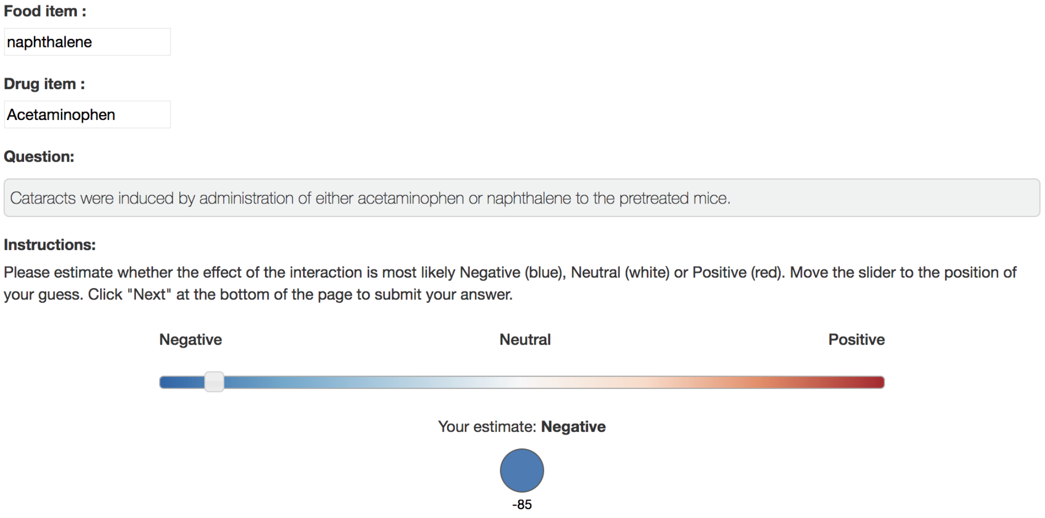


(a) Negative

(b) Neutral

(c) Positive

Figure 5: **Example of Slider Element** The slider element used for classifying examples in a position representing each classification.



Food item :

naphthalene

Drug item :

Acetaminophen

Question:

Cataracts were induced by administration of either acetaminophen or naphthalene to the pretreated mice.

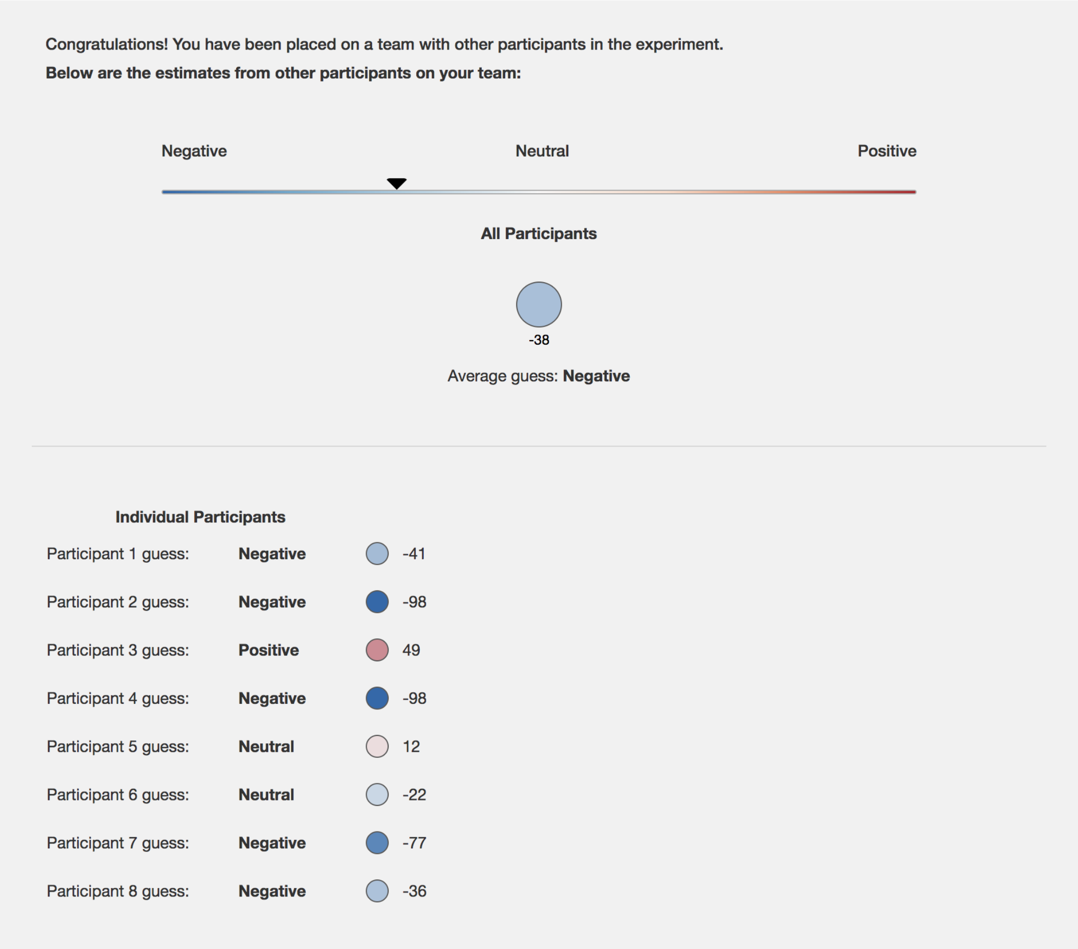
Instructions:

Please estimate whether the effect of the interaction is most likely Negative (blue), Neutral (white) or Positive (red). Move the slider to the position of your guess. Click "Next" at the bottom of the page to submit your answer.

Your estimate: **Negative**

-85

Figure 6: **Example of Baseline** A baseline question with slider in the negative position.



Congratulations! You have been placed on a team with other participants in the experiment.

Below are the estimates from other participants on your team:

All Participants

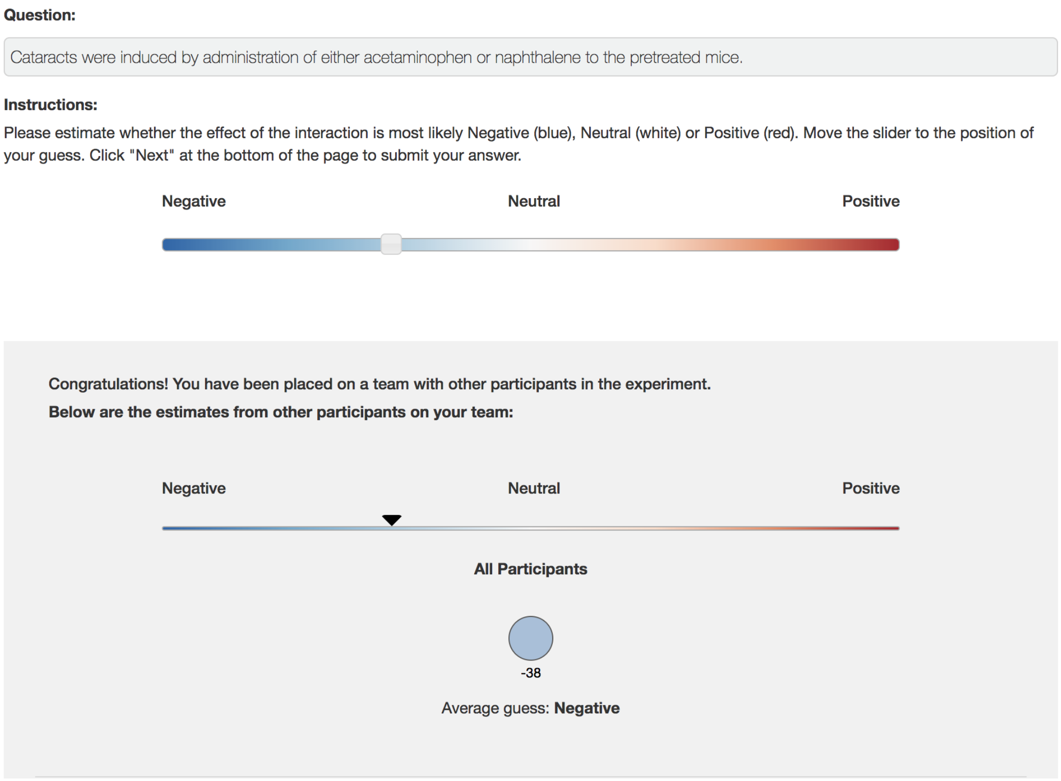
-38

Average guess: **Negative**

Individual Participants

Participant 1 guess:	**Negative**	●	-41
Participant 2 guess:	**Negative**	●	-98
Participant 3 guess:	**Positive**	●	49
Participant 4 guess:	**Negative**	●	-98
Participant 5 guess:	**Neutral**	○	12
Participant 6 guess:	**Neutral**	○	-22
Participant 7 guess:	**Negative**	●	-77
Participant 8 guess:	**Negative**	●	-36

Figure 7: **Example of Treatment** The network information from the previous layer presented as treatment.



Question:

Cataracts were induced by administration of either acetaminophen or naphthalene to the pretreated mice.

Instructions:

Please estimate whether the effect of the interaction is most likely Negative (blue), Neutral (white) or Positive (red). Move the slider to the position of your guess. Click "Next" at the bottom of the page to submit your answer.

Congratulations! You have been placed on a team with other participants in the experiment.

Below are the estimates from other participants on your team:

Average guess: **Negative**

Figure 8: **Example of Treatment Delivery** Network information is propagated to the next layer and implemented as default slider position (e.g. the previous mean estimate).