Valuable prizes!

- The prizes go to Bálint Turi-Kováts, Botond Kiss, and József Pintér
- There were several good solutions (in no particular order): Marcell Nagy, Mátyás Tarnay, Attila Ragács, Nóra Stumphauser, Tamás Nagy, Máté Büki, Adél Juhász, András Simon, Katalin Ócsai, Klaudia Kompis, Tasci Zeynap, Márton Veres
- And several other whose work makes us think they could do this: Alexandra Gerner, Bence Lestyán, Melinda Juhász, Enikő Polyák, Péter Kovács, Tamás Havas
- Others (not listed here to protect the guilty) need to work on this, and nothing at all came from Bálint Király and Dániel Keliger (this is not acceptable)

Methods used so far

- Gradient Boosted Trees
- Decision Trees
- Logistic Regression
- Linear Regression
- Random Forest
- feedforward NN
- k-Nearest Neighbor

To go further with this

1. Typical errors (e.g. testing on first 10 utterances, rather than all utterances of first 4+4+2 speakers) need to be fixed
2. Use your existing method(s) to train on all or just the unstarred, test on all or just the unstarred, report all 4 figures
3. While you are doing this, we will create a leaderboard and maybe a joint github repo? Your work should be named HW1_XY.ipynb
4. Everybody please use jupyter/colab infrastructure
5. Everybody please get an ID on slack (or tell us about their current ID) so that we can invite them to the #aml20 channel on Deep Learning Önképzőkör
6. We will use the slack to discuss the "English or Hungarian" issue raised by a student. The opinion of those higher on the leaderboard will matter more!

The boring linear stuff

1. We start covering this in the next 15-30 minutes, while people fix their stuff and send in the results (please use AML2020 in the Subject: of your msg)
2. We will begin with `highleyman_1962.pdf`
3. Some terminological changes in the past 60 years: *receptor* $\rightarrow$ *feature extractor*; *measurement space* $\rightarrow$ *feature space*; etc.

A significant quote

Even if the optimal decision function were known, its implementation would require, in general, the use of a digital computer or other complex equipment. The cost of such equipiment may, in many cases, outweigh the advantages of mechanized categorization.

What are the issues?

- Categories are numbered $1, 2, \ldots, p$, 0 reserved for *reject*, occurrence probabilities ω_i.
- $\beta(m|i)$ the probability of measurment m from true class i, generally not known, often assumed (n-dim) Gaussian
- $\delta(d_j|m)$ is the classifier making decision d_j given input m
- $C_{i,j}$ is the cost of making decision d_j given input from class i, $C_{ij} > C_{i0} > c_{ii} (= 0)$
- Expected loss $C(\delta) = \sum_{ij} c_{ij} \omega_i \beta(m|i) \delta(d_j|m) dm$
- Chow (1957) proved that indicator functions (1 at a region, 0 elsewhere) are the best δ, optimum decision function depends only on $\omega_i \beta(m, i)$
- The notion of a *decision boundary* is introduced, we are looking for approximating these by hyperplanes given by $\langle \alpha, x \rangle + \alpha_0 \geq 0$

OK boomer (actually, silent gen)

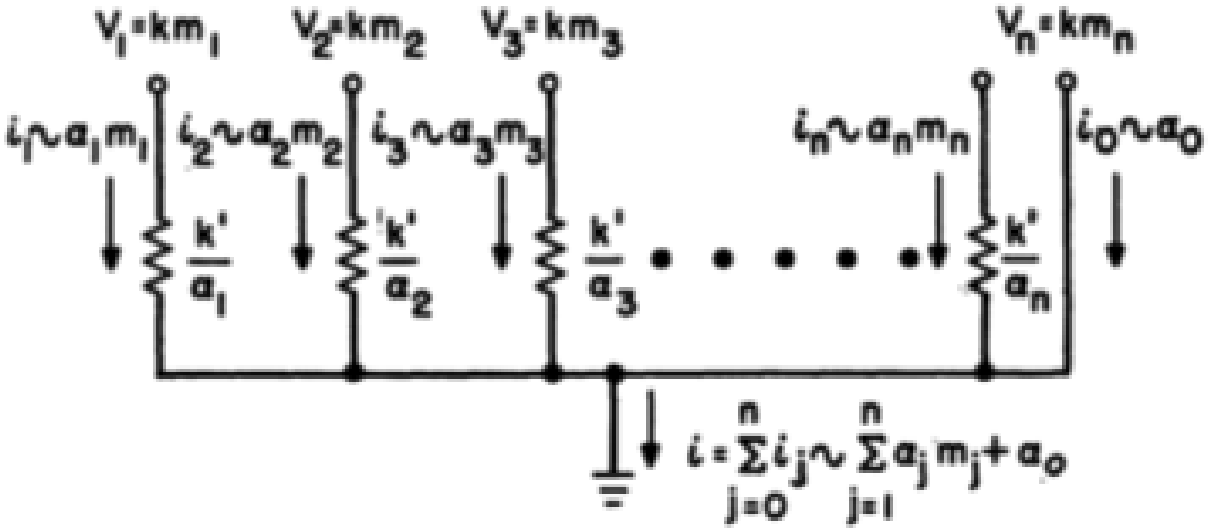

Fig. 2—Implementation of a hyperplane.

The money quote

In order to classify a point m (that is, to recognize an input pattern), it is only necessary to evaluate quantities like $\langle \alpha, x \rangle + \alpha_0$. But such a calculation can be done with several varieties of very inexpensive networks, such as the resistive adder shown in Fig. 2.

Prototype-based classification

- Assume we have k_i samples $m_{i1}, \ldots, m_{ik_i}$ from class i
- We are looking for a *prototype* M_i from each class, which will be our model
- Classification is performed by comparing a new sample m to each of the models M_i, and choosing the one with the best fit
- The classifier is characterized by the M_i and the fit (distance) function d
- Assume everything is linear, and d is euclidean distance, how do we select the best M_i
- First, let's implement a prototype model where the M_i are just the average of the $m_{i,}$ samples.
- Can this be improved? How? This will be the 1st homework!

1st homework

- Read Klautau (2002). Weka people can dig into the details
- Start stealing good ideas from the best notebooks! Everybody needs to call their own HW1_XY.ipynb, where XY are the first two letters of the authur's Neptun code. When you just took someone else's code, give credit in comments. (If you substantially reworked it, but the idea wasn't original with you, still give credit.)
- Your 1st homework is to build the best prototype classifier for the Peterson-Barney data, and comparing it to the (typically more advanced) method you used before under all four conditions (all train/all test; nonstarred train/all test; all train/nonstarred test; nonstarred train/nonstarred test)
- You can perform any kind of data manipulation you like, as long as your model is truly a prototye-based model
- This homework is serious, it **will** be part of your grade. Due Monday night the absolute latest

Project discussion

- Read Highleyman (1962) further. Maybe you want to present the stuff in class (this is one kind of project)?
- Think of a problem domain: ASR; OCR; biometric identification; pattern classification; ranking/recommendation; info extraction; info retrieval; natural language processing (NLP); financial; medical... You can Bring Your Own Data
- Beating SOTA on any standard task guarantees an A in this course (doesn't guarantee publication these days, but it's a good step)
- First three on the leaderboard will get As (others can too)
- Initial project plans due by 3rd week, if you can't come up with a plan a project will be assigned to you