5. Vehicle Platooning I

金力 Li Jin
li.jin@sjtu.edu.cn
上海交通大学密西根学院
Shanghai Jiao Tong University UM Joint Institute

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Recap

- Single-vehicle on an empty road
  - 1 dimensional
  - 2 dimensional
- Multi-vehicle planning
  - Policy-free approach
  - Policy-based approach

Outline

- Technological basis
  - Autonomous driving
  - Vehicle-to-vehicle coordination
- Classical approach
  - Modeling
  - Decision making
- Learning-based approach
  - Objective
  - Design

Platooning

- https://v.qq.com/x/page/i0877glx6c7.html
- https://v.youku.com/v_show/id_XNDQzNTQ2OTMzNg==

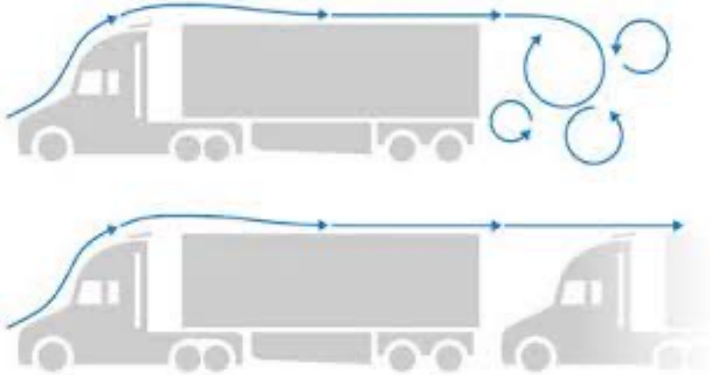
Platooning: Motivation

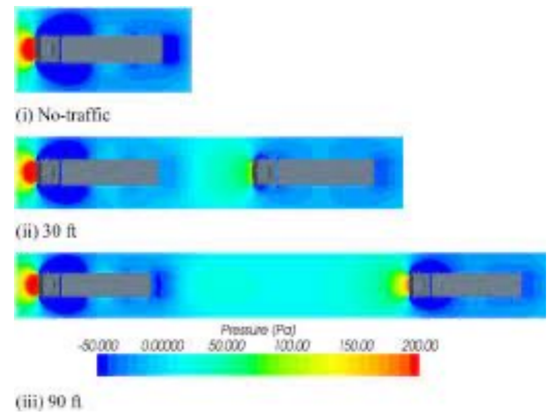
- Limited highway capacity -> more traffic jams
- A naïve solution: build wider roads.
- A smarter solution: reduce the intervehicle distance.

- Before the age of AI, human could not do that.
- But now, computers can do that! (time gap: 2s ->0.5s)

Platooning: Motivation

- Reduces air drag; saves fuel (up to 15%)





- Improved safety & working condition for drivers

Technological basis

- Cooperative adaptive cruise control
- Autonomous driving
- Vehicle-to-vehicle coordination

Cooperative adaptive cruise control

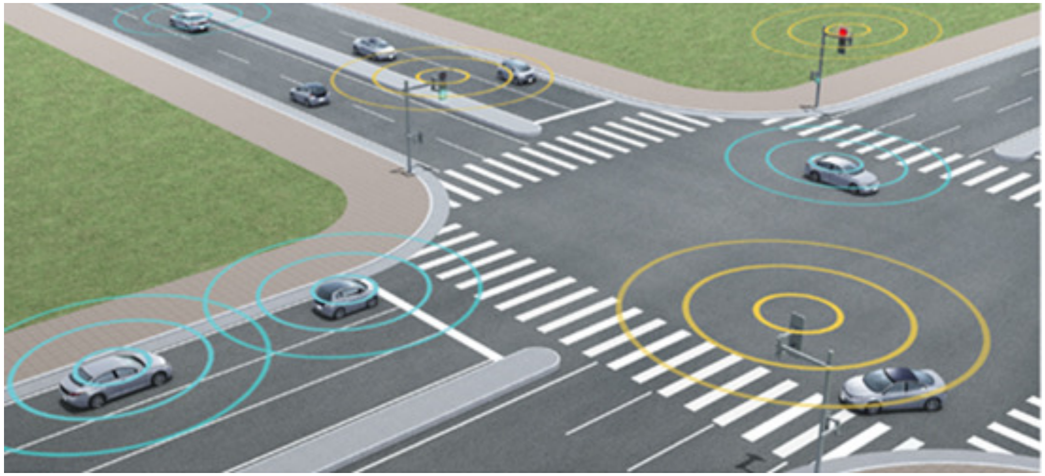
- CACC: 协同自适应巡航控制
- Two key words:
  - Cooperative（协同的）: multiple vehicles share information and jointly make decisions
  - Adaptive（自适应的）: control inputs are generated in response to real-time condition
- CACC drives better than human, since
  - vehicles talk to each other and can proactively and anticipatorily account for the behavior of other vehicles;
  - computers can respond faster than human
- Note: current platooning technology typically requires the leading vehicle to be human-driven. (Why?)

Autonomous driving

- Longitudinal: vehicle following
- Recall lecture 3
- Lateral: mostly lane keeping; sometimes lane changing

Vehicle-to-vehicle coordination

- Onboard unit (OBU)
- Broadcast information to neighboring vehicles:
  - Vehicle type
  - **Latest** position, speed, acceleration, orientation…
  - **Intended** position, speed, acceleration, orientation…

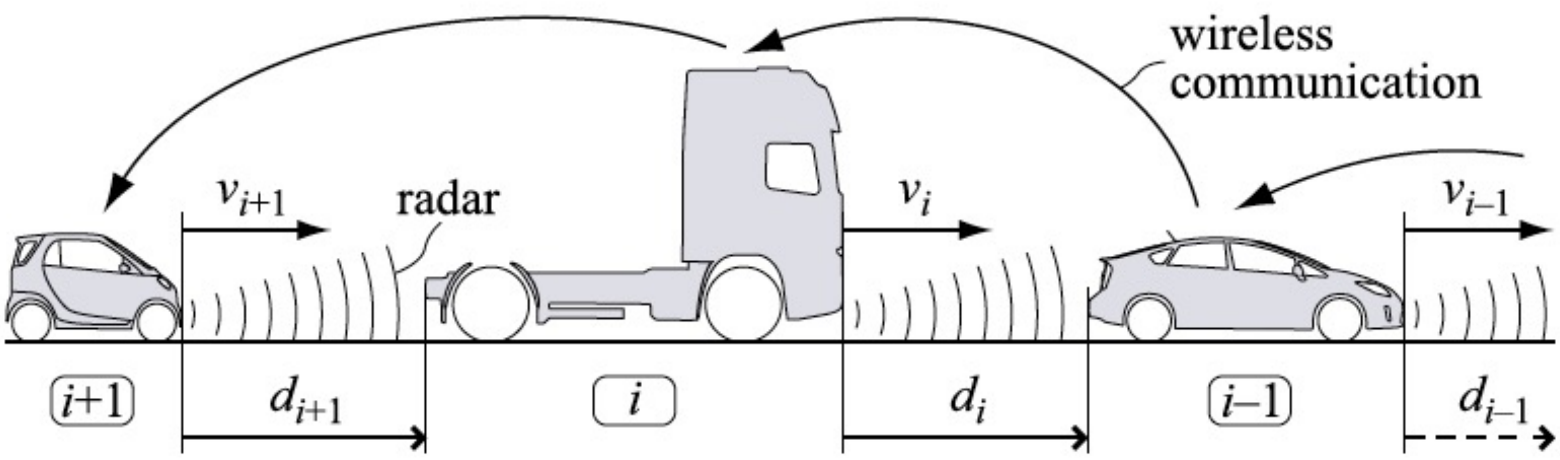


Classical Approach

- Ploeg, J., Van De Wouw, N., & Nijmeijer, H. (2013). L_p string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, *22*(2), 786-793.

Platoon dynamics

- Consider a platoon of m vehicles
- d_i = the distance between vehicle i and its preceding vehicle $i - 1$
- v_i its velocity.
- The objective of each vehicle is to follow the preceding vehicle at a desired distance $d_{r,i}$



Platoon dynamics: Spacing policy

- The objective of each vehicle is to follow the preceding vehicle at a desired distance $d_{r,i}$

$$d_{r,i}(t) = r_i + hv_i(t), \qquad i \in S_m$$

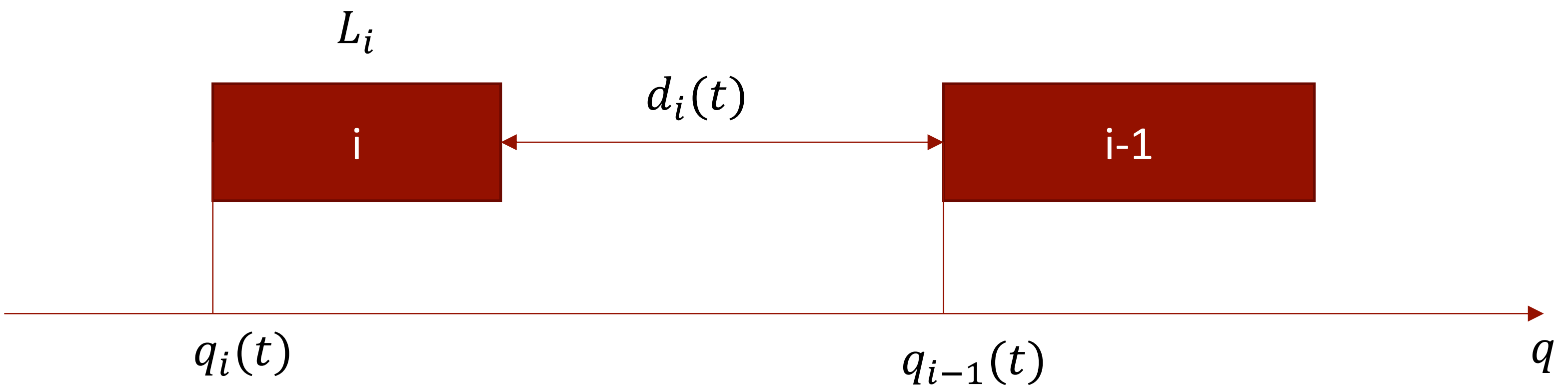
- h = the time headway (assuming homogeneous platoon)
- r_i = the standstill distance.
- $S_m = \{i \in N \mid 1 \leq i \leq m\}$ is the set of all vehicles in a platoon of length $m \in \mathbb{N}$.
- Note: $d_{r,i}$ is a spacing policy that specifies the desired spacing
- This particular controller is nominally stable: i.e., stable if perfectly implemented.
- Spacing policy is easier to design, since it is purely kinematic (运动学的).

Platoon dynamics: Spacing policy

- Spacing error

$$e_i(t) = d_i(t) - d_{r,i}(t)$$
$$= (q_{i-1}(t) - q_i(t) - L_i) - \big(r_i + hv_i(t)\big)$$

- q_i = rear-bumper（后保险杠） spacing of vehicle i
- L_i = length of vehicle i



- Control objective: $\lim_{t\to\infty} e_i(t) = 0$ for all i

Platoon dynamics: Vehicle model

- For each vehicle i

$$\begin{bmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{bmatrix} = \begin{bmatrix} v_{i-1} - v_i \\ a_i \\ -\frac{1}{\tau} a_i + \frac{1}{\tau} u_i \end{bmatrix}$$

- d_i = inter-vehicle spacing
- v_i = vehicle speed
- a_i = vehicle acceleration
- u_i = control input (desired acceleration)
- τ = time constant associated with driveline (传动) dynamics

Platoon dynamics: Vehicle model*

- Ploeg et al. proposed a control law for u_i

$$h\dot{u}_i = -u_i + \begin{bmatrix} k_p & k_d & k_{dd} \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{bmatrix} + u_{i-1}$$

- Dynamic equation for feedback-controlled system:

$$\begin{pmatrix} \dot{e}_i \\ \dot{v}_i \\ \dot{a}_i \\ \dot{u}_i \end{pmatrix} = \begin{pmatrix} 0 & -1 & -h & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\tau} & \frac{1}{\tau} \\ \frac{k_p}{h} & -\frac{k_d}{h} & -k_d - \frac{k_{dd}(\tau-h)}{h\tau} & -\frac{k_{dd}h+\tau}{h\tau} \end{pmatrix} \begin{pmatrix} e_i \\ v_i \\ a_i \\ u_i \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{k_d}{h} & \frac{k_{dd}}{h} & \frac{1}{h} \end{pmatrix} \begin{pmatrix} e_{i-1} \\ v_{i-1} \\ a_{i-1} \\ u_{i-1} \end{pmatrix} \quad (7)$$

Platoon dynamics: Vehicle model

- The dynamic equation can be compactly written as
$$\dot{x}_i = A_0 x_o + A_1 x_{i-1}$$
- $x_i = [e_i\ v_i\ a_i\ u_i]^T$
- A_0 and A_1 defined accordingly
- For vehicle 1, it follows a virtual reference vehicle 0 such that
$$x_0 = \begin{bmatrix} e_0 \\ v_0 \\ a_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{v}_0 \\ 0 \\ 0 \end{bmatrix}$$
- $\bar{v}_0$ is the target & equilibrium speed of the platoon

Platoon dynamics: Asymptotic stability

- Recall: platoon is asymptotically stable if distance error approaches 0, i.e.

$$\lim_{t\to\infty} e_i(t) = 0 \ \ \forall i$$

- Ploeg et al. proved that if the controller satisfies

$$k_p > 0, k_d > 0, k_{dd} > 1, (1 + k_{dd})k_d > k_p\tau,$$

then the platoon is asymptotically stable.

- Asymptotic stability: with constant reference speed & no disturbances, platoon will gradually converge to equilibrium state.
- What if there is disturbances?
- String stability: disturbances are not amplified as they propagate in the platoon. #

Platoon dynamics: String stability*

- How do we know whether disturbances are amplified in a dynamical system? -> Frequency domain response.
- Consider a linear time-invariant (LTI) single-input-single-output (SISO) system

$$\dot{x} = ax + bu$$
$$y = cx + du$$

- Suppose the input is sinusoidal

$$u(t) = \bar{u}e^{j\omega t}$$

- For LTI SISO systems, the output is also sinusoidal

$$y(t) = \bar{y}e^{j\omega t + \phi}$$

- Amplification is characterized by $\frac{\bar{y}}{\bar{u}}$

Platoon dynamics: String stability*

- $\frac{\bar{y}}{\bar{u}}$ determines whether disturbances are amplified or suppressed
  - If $\frac{\bar{y}}{\bar{u}} > 1$, disturbances are amplified, and the system will blow up.
  - If $\frac{\bar{y}}{\bar{u}} < 1$, disturbances are damped, and the system will converge.
- Mathematically, $\frac{\bar{y}}{\bar{u}}$ is equal to the magnitude of the system's frequency response function (FRF) $\Gamma(j\omega)$

$$\frac{\bar{y}}{\bar{u}} \leq \sup_{\omega>0} |\Gamma(j\omega)|$$

- For more information on frequency response functions, see e.g. Oppenheim, Alan, and George Verghese. *Signals, Systems and Inference*. Prentice Hall, 2015. ISBN: 9780133943283.

Platoon dynamics: String stability*

- Now let's go back to vehicle platooning
- Each vehicle can be modeled by an FRF $\Gamma_i(j\omega)$
- Then, vehicle i damps disturbance if
$$\sup_{\omega>0}|\Gamma_i(j\omega)| < 1$$
- A platoon is a cascaded system
$$\dot{x}_i = A_0 x_o + A_1 x_{i-1}$$
- Therefore, any disturbance will get damped as it propagates through the platoon if
$$\sup_{\omega>0}|\Gamma_i(j\omega)| < 1 \quad \text{for all } i$$
- The above property is called string stability.

Learning-based approach

- Neural networks
- Learning-based adaptive control

Online model identification

- Recall that the classical control synthesis was based on the linear model

$$\begin{bmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{bmatrix} = \begin{bmatrix} v_{i-1} - v_i \\ a_i \\ -\frac{1}{\tau} a_i + \frac{1}{\tau} u_i \end{bmatrix}$$

- The controller was also given by linear ODE (PD control*)

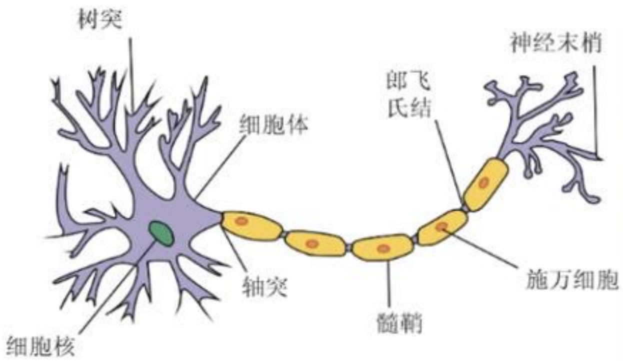
$$h\dot{u}_i = -u_i + [k_p \; k_d \; k_{dd}] \begin{bmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{bmatrix} + u_{i-1}$$

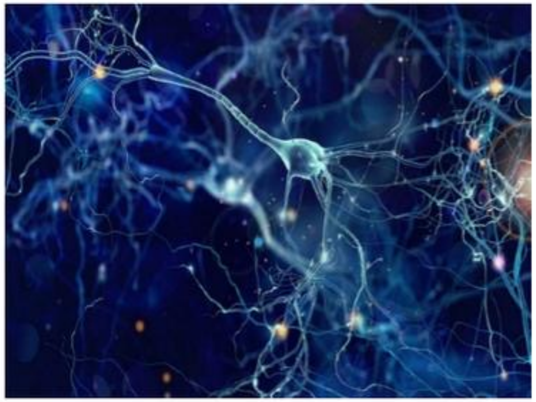
- In practice, such linearization may not be perfect:
  - How do you know that the system is perfectly linear? What if nonlinear? (For example, if τ is state dependent.)
  - How do you know that a linear controller is sufficiently good? What if a nonlinear controller can do much better than a linear one?

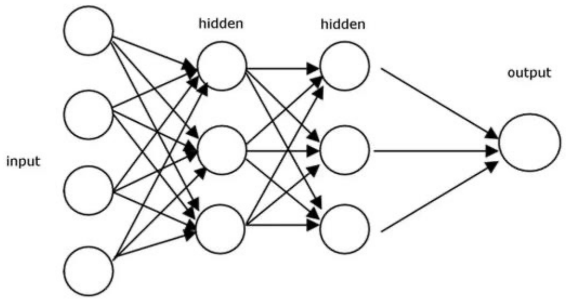
Learning-based approach: Neural networks

A simple but usually useful solution: use neural networks to approximate the dynamics or the controller!

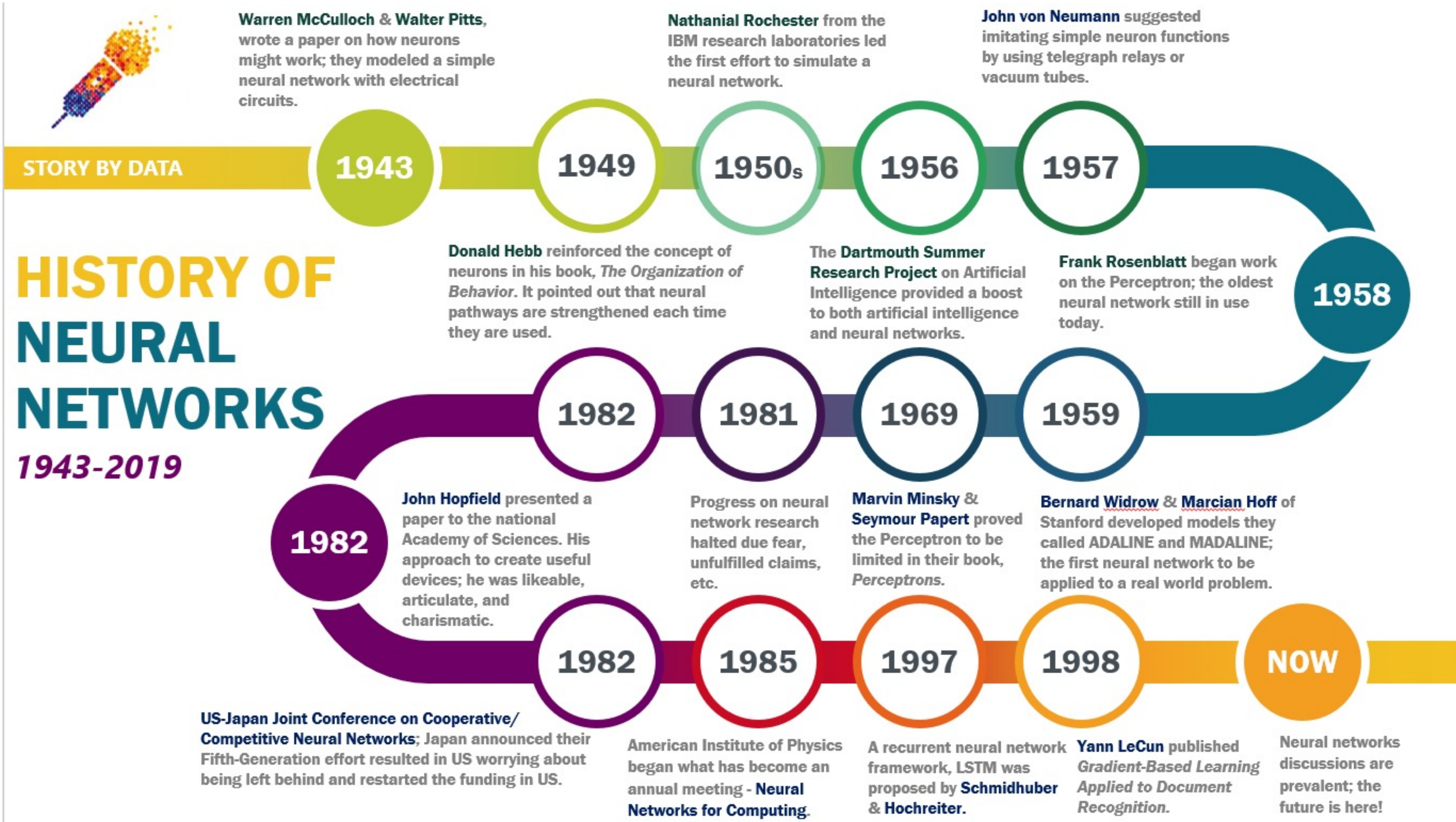
- Artificial neural networks have displayed promising performance and flexibility in other domain. characterized by high degrees of noise and variability
- Neural networks are nothing but a class of functions that provide strong approximation capability.







History



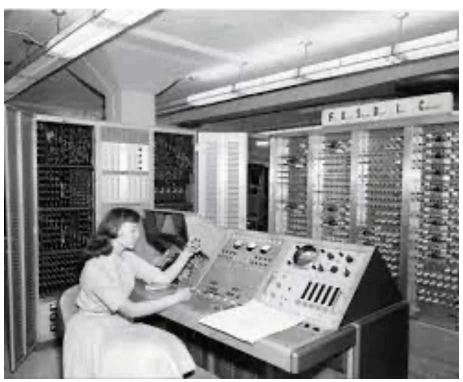
https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec

Question

Why were neural networks not hot until quite recently?

Evolution of computers

Evolution of data technology



Evolution of communication technology

Basic idea

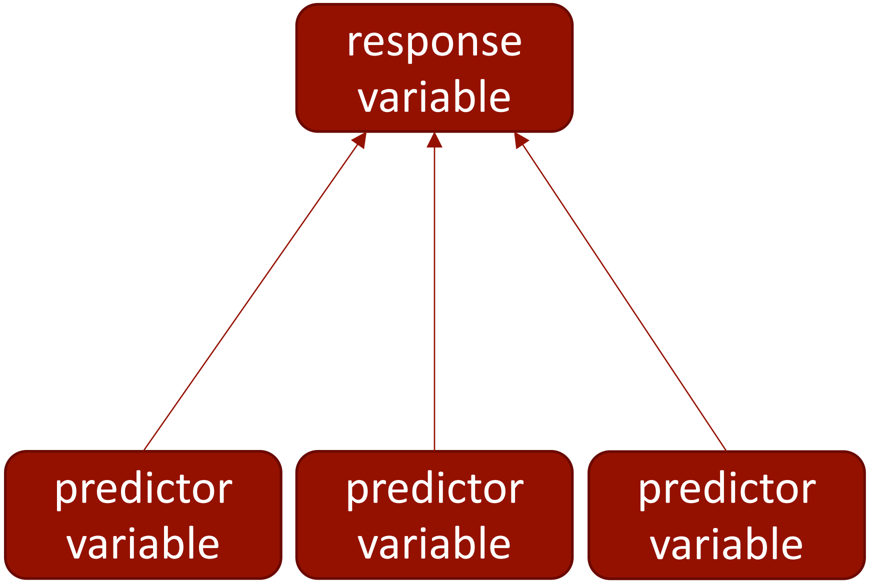
- A class of learning methods that was developed separately in different fields—statistics and artificial intelligence—based on essentially identical models.
- The central idea is to extract linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features.
- The result is a powerful learning method, with widespread applications in many fields.
- Used in everywhere in smart cities: autonomous driving, intelligent transportation systems, smart grids, urban informatics…

Background

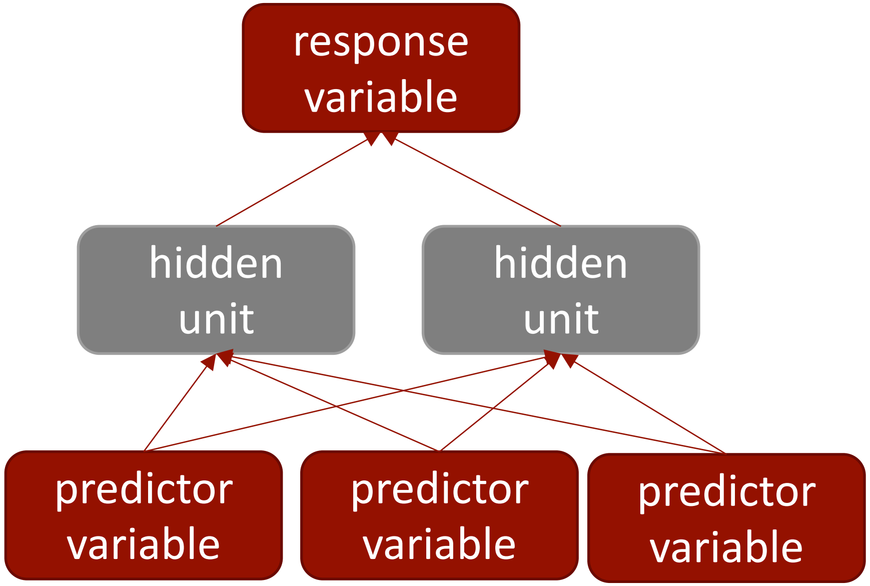
- The term neural network has evolved to encompass a large class of models and learning methods.
- Here we describe the most widely used neural net, sometimes called the single hidden layer back-propagation network, or single layer perceptron.
- There has been a great deal of hype surrounding neural networks, making them seem magical and mysterious.
- As we make clear in this course, they are just nonlinear statistical models.

Introduction

- A two-stage regression or classification model
- Instead of directly feeding predictor variables into a regression/classification function, we put a set of intermediate derived features or hidden units in between.



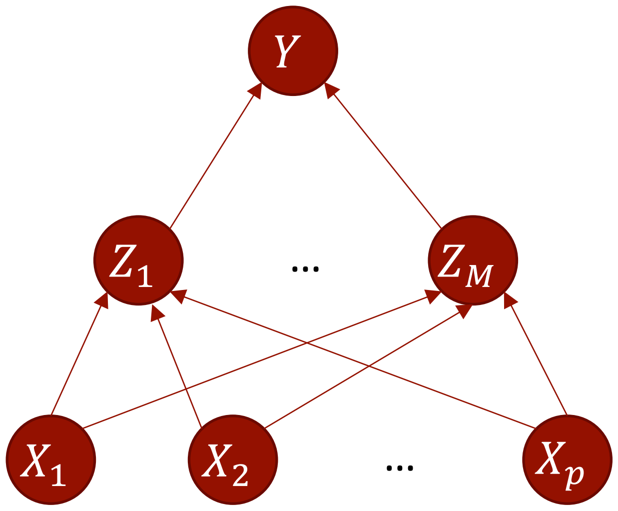
linear regression (LR)



neural network (NN)

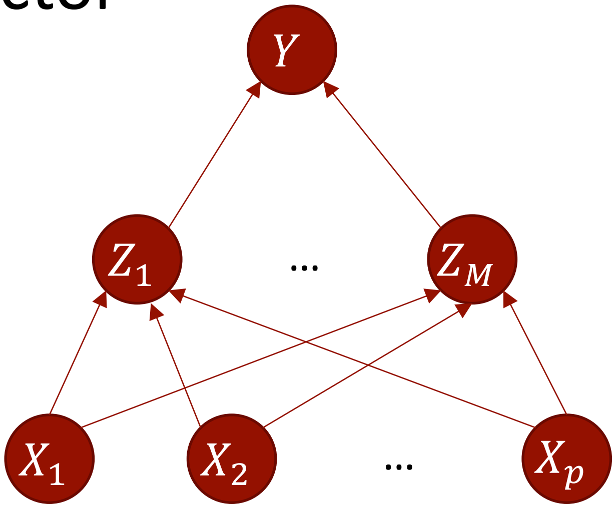
NN regression: 1-dimensional output

- Suppose we have a p-dimensional input vector $X = [X_1\ X_2\ \ldots X_p]^T$
- Our objective is to predict an output scalar Y from X
- Hidden units: an M-dimensional vector $Z = [Z_1\ Z_2\ \ldots\ Z_M]^T$
- Z is given by the sigmoid function:
- $Z_1 = \frac{1}{1+\exp(\alpha_{01}+\alpha_1^T X)}$
- $Z_2 = \frac{1}{1+\exp(\alpha_{02}+\alpha_2^T X)}$
- $Z_m = \frac{1}{1+\exp(\alpha_{0m}+\alpha_m^T X)}$
- $m = 1,2, \ldots, M$

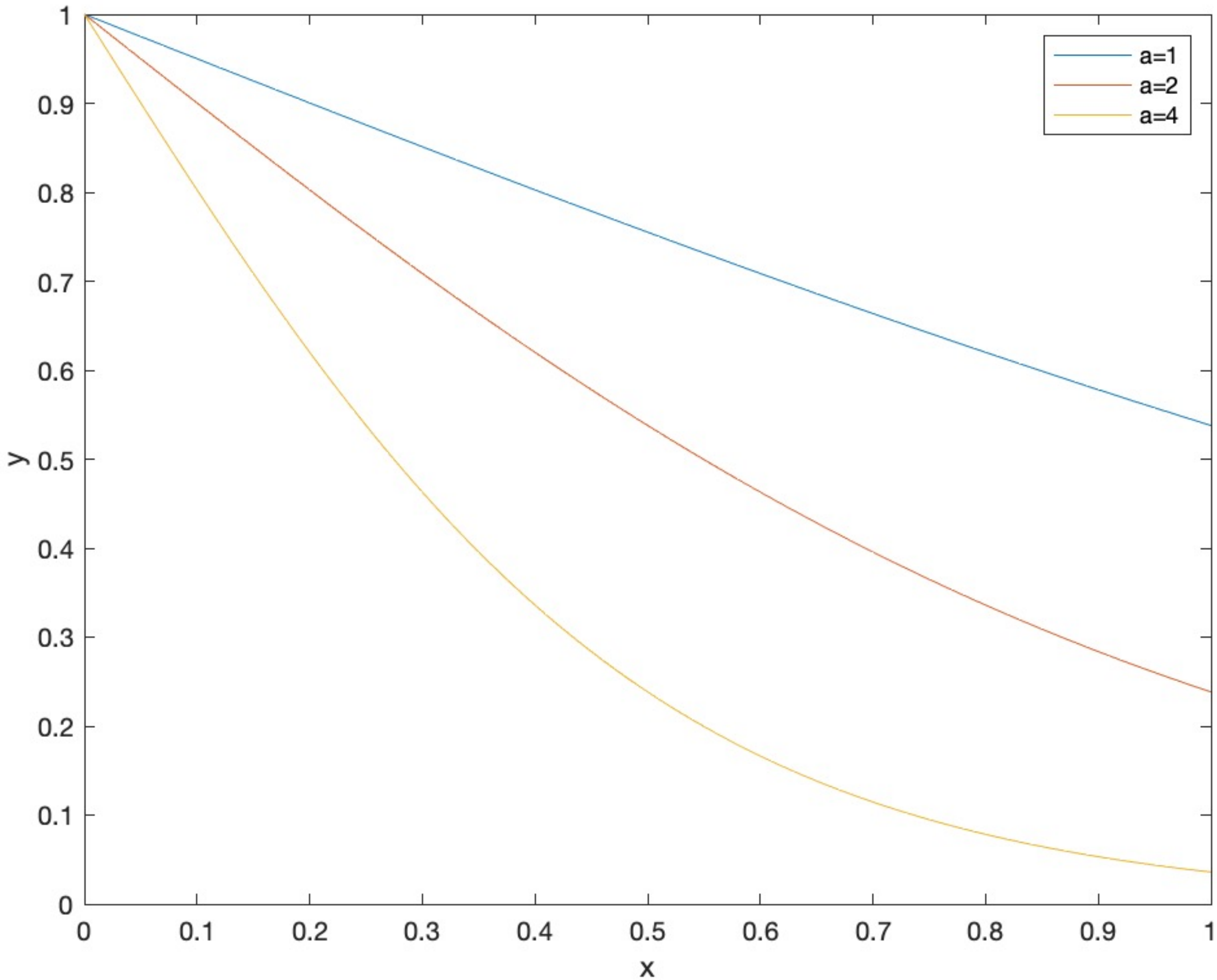


NN regression: 1-dimensional output

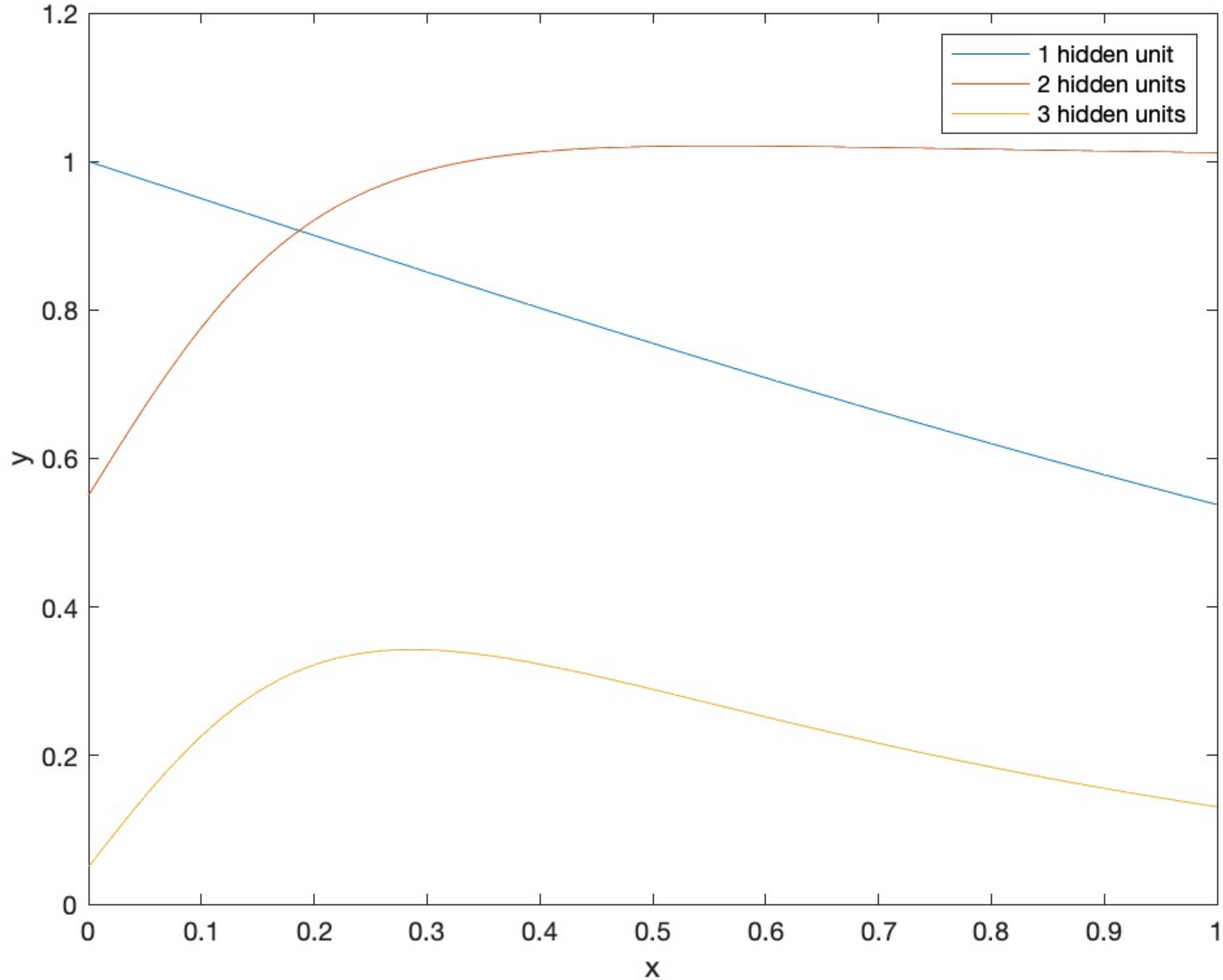
- $Z_m = \frac{1}{1+\exp(\alpha_{0m}+\alpha_m^T X)}, \ m = 1,2,\ldots,M$
- α_{0m} is a scalar
- α_m is a p-dimensional vector
- Output $Y = \beta_0 + \beta^T Z$
- β_0 is scalar, β is M-dimensional vector
- Thus, we have constructed a neural network.
- The NN is essentially a nonlinear regression.



Example: 1 input, 1 hidden unit



Example: 1 input, 1--3 hidden units



Hidden units

- The units in the middle of the network, computing the derived features Z_m, are called hidden units because the values Z_m are not directly observed.
- In general there can be more than one hidden layer: deep neural networks.
- We can think of the Z_m as a basis expansion of the original inputs X; the neural network is then a standard linear model, or linear multilogit model, using these transformations as inputs.
- There is, however, an important enhancement over the standard basis expansion techniques discussed; here the parameters of the basis functions are learned from the data.

Hidden units

- General form of hidden unit:

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X)$$

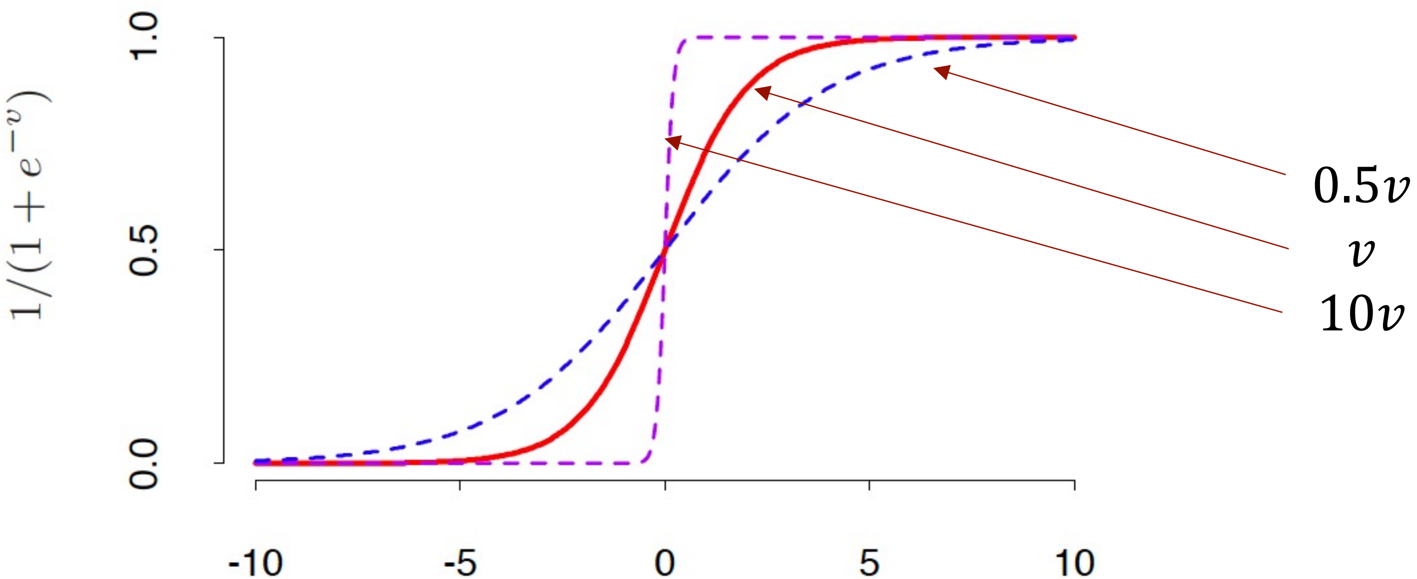
- Notice that if σ is the identity function, then the entire model collapses to a linear model in the inputs.
- Hence a neural network can be thought of as a nonlinear generalization of the linear model, both for regression and classification.
- By introducing the nonlinear transformation σ, it greatly enlarges the class of linear models.
- Typical choice: sigmoid function

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X) = \frac{1}{1 + \exp(\alpha_{0m} + \alpha_m^T X)}$$

Impact of α_m

$$Z_m = \frac{1}{1 + \exp(\alpha_{0m} + \alpha_m^T X)}$$

- The rate of activation of the sigmoid depends on the norm of α_m , and if $||\alpha_m||$ is very small, the unit will indeed be operating in the linear part of its activation function.



Learning-based adaptive control

- Let’s look at the two objects to be approximated:
  - Dynamics
  - Controller
- Note: both of the above are nothing but mappings (functions). Hence, we can approximate them using neural networks.
- Both of the above are in the form of ODEs:

$$\dot{\xi}(t) = f\big(\xi(t)\big)$$

- Input: state variables $\xi(t)$ on the right
- Output: time derivatives $\dot{\xi}(t)$ on the left
- Train an NN $\hat{f}$ by minimizing $\sum_{s=0}^{t}\left\|\dot{\xi}(s) - \hat{f}\big(\xi(s)\big)\right\|_2^2$

Learning-based adaptive control

- Dynamics

$$\begin{bmatrix} d_i \\ v_i \\ a_i \\ u_i \\ v_{i-1} \end{bmatrix} \rightarrow NN \rightarrow \begin{bmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{bmatrix} \text{ instead of } \begin{bmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{bmatrix} = \begin{bmatrix} v_{i-1} - v_i \\ a_i \\ -\frac{1}{\tau} a_i + \frac{1}{\tau} u_i \end{bmatrix}$$

- Controller

$$\begin{bmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \\ u_i \\ u_{i-1} \end{bmatrix} \rightarrow NN \rightarrow \dot{u}_i \text{ instead of } h\dot{u}_i = -u_i + \begin{bmatrix} k_p & k_d & k_{dd} \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{bmatrix} + u_{i-1}$$

Learning-based adaptive control

- In fact, the NN can be updated in an online manner.
- Suppose that we have no prior knowledge about the system dynamics.
- We want to learn the dynamics $\dot{\xi} = f(\xi)$
- We begin with an initial guess $\hat{f}^0(\xi)$, i.e. an NN
- At time t, we update our estimate via

$$\min \sum_{s=0}^{t} \left\| \dot{\xi}(s) - \hat{f}^t\big(\xi(s)\big) \right\|_2^2$$

- Thus, $\hat{f}^t$ should converge to the “true” dynamics as t increases.

Learning-based adaptive control

- Such an approach is adaptive in the sense that it can track non-stationarity.
- We begin with an initial model $\hat{f}^0(\xi)$, i.e. an NN
- At time t, we update our estimate via

$$\min \sum_{s=0}^{t} \rho^{t-s} \left\| \dot{\xi}(s) - \hat{f}^t\big(\xi(s)\big) \right\|_2^2$$

- Thus, if the system dynamics varies over time, $\hat{f}^t$ should track the time-variant dynamics as t increases.
- For example, a learning-based adaptive driver can adjust itself in response to changes in # of passengers, weather, road surface, etc.

Course project #1: vehicle platooning

- Study the performance of a CACC algorithm in Ploeg, J., Van De Wouw, N., & Nijmeijer, H. (2013). L_p string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, *22*(2), 786-793.
- Good: replicate the results in Section VI and explore further a little bit.
- Excellent: add learning-based adaptivity to the controller.

Summary

- Technological basis
  - Autonomous driving
  - Vehicle-to-vehicle coordination
- Classical approach
  - Modeling
  - Decision making
- Learning-based approach
  - Objective
  - Design

Next time

- Background
  - Connected & autonomous vehicles
  - Vehicle-to-infrastructure connectivity
- Control in nominal case
  - Centralized approach
  - Decentralized approach
  - Hierarchical control
  - **HW2**
- Control in face of disruptions
  - How to address latency
  - How to address packet loss
  - How to address malicious attacks