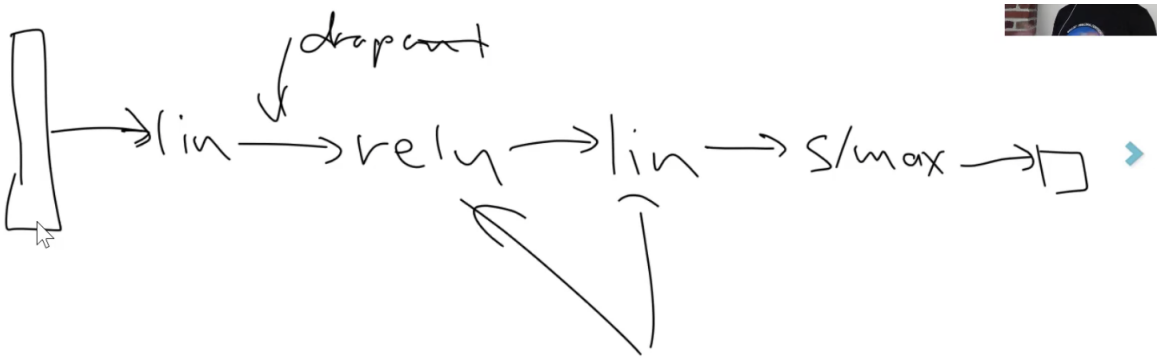
structured data & entity embeddings

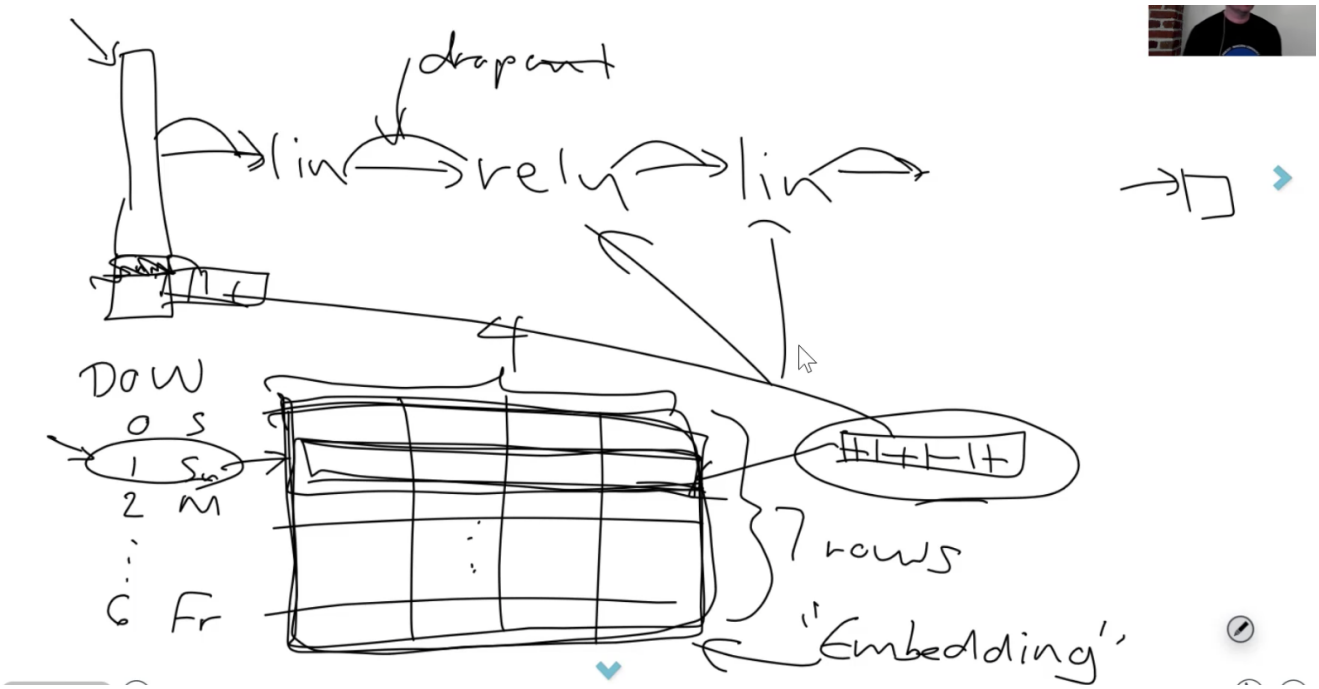
- if you have a column of features and are trying to predict something like

Date	Store	DayOfWeek	Year	Month	Day	StateHoliday	CompetitionMonthsOpen	Promo2Weeks	StoreType
2014-01-08	781	3	2014	1	8	False	24	0	
2014-11-22	626	6	2014	11	22	False	12	0	

- categorical variables should be handled as categorical variables. we shouldn't encode it as 0,1,2 etc because there might not be a pattern based on size of number
- continuous variables can either be modeled as continuous or as categorical depending on if you are sure magnitude of number means something → this is a modeling decision you have to make.
- we can preprocess continuous variables to have mean 0 and standard deviation 1 → neural networks really like this format.
- then just feed in the feature vector as input with target variable as output using a feedforward fully connected neural net



- categorical variables can be fed in as entity embedding by looking up their embedding in a embedding matrix. if you encode categorical features in a one-hot fashion, then the weights of the next layer can act as the embedding matrix
- how do you train this entity embedding? with a suitable method on a suitable task. like we do for word2vec.



- if there are categorical variables, add a special feature for unknown/unspecified data to prepare to make classifications/regression even with missing values. so if there are 7 days in a week, we'll use a length 8 one hot vector for the day of the week category.
- eg instacart and pinterest have embedding matrices for products, stores etc → entity embeddings
- these embedding vectors tend to get rich semantic behavior → like properties of a sunday (most ppl aren't working, might be preparing for work week etc)
- so if you have the option of making an embedding categorical, make it categorical so that you can learn an embedding and get all this rich semantic information (assuming you have enough data and come up with an unsupervised training objective)
- this kind of entity embedding + deep learning has helped lots of companies in the valley get state of the art recommender performance with much lesser feature engineering and maintenance than random forests, gradient boosting machines etc