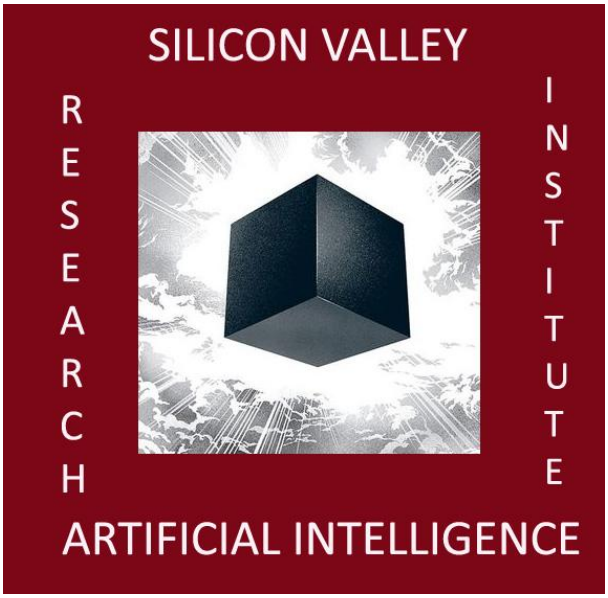

Neural Networks as Vector Spaces

When a neural network is trained to "learn" some pattern, its neurons get organized in a rather geometric manner. A neural network basically constructs a high-dimensional space in which the distance between two points mirrors the degree of relationship between two objects in the real world. For example, two words that tend to show up together in many sentences will be represented by two points very close to each other. These "points" in high-dimensional spaces are "vectors". This was the discovery made by Tomas Mikolov's team at Google in 2013, using the "skip-gram" method for constructing vector representations of words from analyzing large sets of text, a method now known as "word2vec" ("Distributed Representations of Words and Phrases and their Compositionality", 2013). The same approach can be used to analyze images or speech, and, again, turn a large set into a high-dimensional vector space. Now you can use an old mathematical tool called "vector arithmetic" and perform calculations on these vectors. Mikolov showed that, for example, one can perform this algebraic operation: "vec(king) - vec(man) + vec(woman)" and obtain "vec(queen)". Unfortunately (or luckily), this method ended up revealing embarrassing biases in the texts of our world. For example, in 2016 Tolga Bolukbasi at Boston University used a neural network to calculate "vec(father) - vec(doctor) + vec(mother)". You would expect the answer to be still "doctor" as there are many female doctors, but instead the answer (obvious once you see it) was "nurse": the line from male doctor to female doctor is not a straight one. As Einstein would put it, the vector space is warped!

The skip-gram model, just like the bag-of-words model, needs to be trained to minimize a "loss function". The problem is that probabilistic models of language like these are computationally prohibitive because they require the mathematical processing of an entire vocabulary, which consists of tens of thousands of words. Two main "short cuts" were in use: "hierarchical softmax", by Yoshua Bengio's student Frederic Morin at the University of Montreal ("Hierarchical Probabilistic Neural Network Language Model", 2005), and "noise-contrastive

estimation", developed by Michael Gutmann and Aapo Hyvarinen at the University of Helsinki ("Noise-contrastive Estimation", 2010). Mikolov used "negative sampling", a simplified variation of noise-contrastive estimation.