In [1]:

```
import numpy as np
import matplotlib.pyplot as plt #would like a better plotting library
import scipy
%matplotlib inline
```

NBconvert - set all headings to add an id so i can link to them

Contents

t - SNE

Definition

Given a dataset of N high-dimensional objects $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathrm{X}$, t-SNE first computes probabilities p_{ij} that are proportional to the similarity of objects $\mathbf{x}_i$ and $\mathbf{x}_j$, as follows:

$$p_{j|i} = \frac{\exp(-\| \mathbf{x}_i - \mathbf{x}_j \|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\| \mathbf{x}_i - \mathbf{x}_k \|^2 / 2\sigma_i^2)} \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

The bandwidth of the Gaussian kernels σ_i, is set in such a way that the perplexity of the conditional distribution equals a predefined perplexity using a binary search. As a result, the bandwidth is adapted to the density of the data: smaller values of \sigma_i are used in denser parts of the data space.

t-SNE aims to learn a d-dimensional map $\mathbf{y}_1, \ldots, \mathbf{y}_N$ (with $\mathbf{y}_i \in \mathrm{R}^d$) that reflects the similarities p_{ij} as well as possible. To this end, it measures similarities q_{ij} between two points in the map $\mathbf{y}_i$ and $\mathbf{y}_j$, using a very similar approach. Specifically, q_{ij} is defined as:

$$\frac{(1 + \| \mathbf{y}_i - \mathbf{y}_j \|^2)^{-1}}{\sum}$$

$$q_{ij} = \frac{(1 + \|\mathbf{y}_k - \mathbf{y}_\ell\|^2)^{-1}}{\sum_{k \neq \ell} (1 + \|\mathbf{y}_k - \mathbf{y}_\ell\|^2)^{-1}}$$

Herein a heavy-tailed Student-t distribution is used to measure similarities between low-dimensional points in order to allow dissimilar objects to be modeled far apart in the map.

The locations of the points \mathbf{y}_i in the map are determined by minimizing the (non-symmetric) Kullback–Leibler divergence of the distribution Q from the distribution P, that is:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The minimization of the Kullback–Leibler divergence with respect to the points $\mathbf{y}_i$ is performed using gradient descent. The result of this optimization is a map that reflects the similarities between the high-dimensional inputs well.

Motivation

To efficiently ???

- Find topology of data? It would be nice to ?

Background

Metric spaces and distances

Perplexity and

need to learn this

In [ ]:

```
```

Gaussians

t

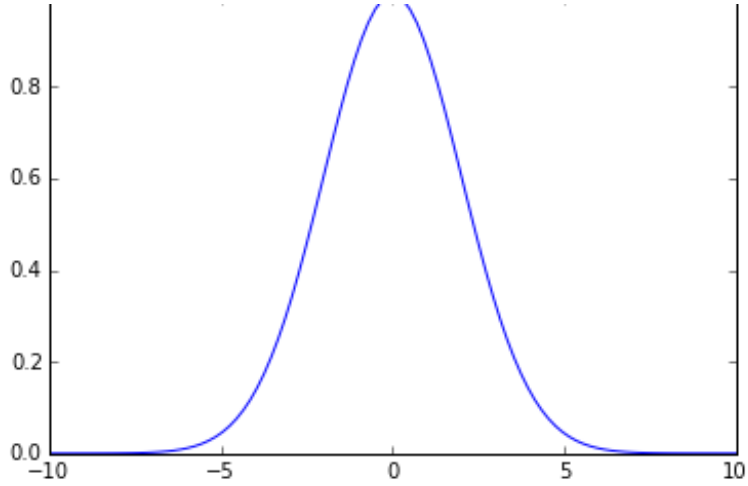
$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

In [2]:

```
x = np.linspace(-10,10,100)
a = 1 #the height
b = 0 #the x/center position
c = 2 #the width
f = a*np.exp(-((x-b)**2)/(2*(c**2)))
plt.plot(x,f)
```

Out[2]:

```
[<matplotlib.lines.Line2D at 0x105b00da0>]
```



Kullback–Leibler divergence

It is also called the relative entropy of P with respect to Q, and written H(P|Q).

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

Pictures to explain would be good

In [ ]:

In [ ]:

Method

Cost function

The cost function and derivates - need to derive this here!

$$\mathbf{C} = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$\frac{\partial \mathbf{C}}{\partial \mathbf{y}} = 2 \sum_j (\mathbf{y}_i - \mathbf{y}_j)(p_{ij} - q_{ij} + p_{ji} - q_{ji})$$

Optimisers

So why does steepest descent not work? it has no way to get past local minima (a picture of the many local minima would be nice). Momentum and annealing (random jitter) help bump the

Like a rock rolling down a hill. The problem with steepest gradient descent is that ~~the gradients are too low and~~ there are too many ledges and bluffs to get caught on. It will instantly stop, regardless of momentum, on any of these. ... Momentum work better sa it will continue on throughSo as soon as it

So I guess the questions this raises are:

So I guess the questions this raises are;

- how can we find the global maximum?
- how can we get out of local minima? /
  - random jumps,
  - tunnel,
  - use the history of the trajectory (momentum/integral control),
- what are the best methods for optimising surfaces with many local minima? (surely whatever the solution it will be expensive?
  - is this the sort of thing particle swarms are good at?
- how can we reduce the amount of local minima in/on the surface?

Need to learn how adagrad, adam, ...

Implementation

In [ ]:

```
```

Results and experiments

To test

- what happens if you init all y at the same position?
- limits as gaussian width tends to 0 and inf

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

Questions and thoughts

- The two functions to find p and q should be the same??
- Slower rate of dimensionalty reduction, discrete, continuious. (reminds me of a recursive function?)
- What if the data set is not a bunch of vectors, but matrixies, or other data structures??
- The method needs to reflect the data somehow?
- Are we only allowing orthogonal dimensions? This seems to be quite limiting?? For example, what if we could have dimensions that are curves? So MNIST would be a small number of curve dimensions??
  - so what if we mapped it into non-euclidean grometry and then back?
    - for example - images: we could mapp the pixel intensities into a line segments/curves and them back to a 2d position vector?
    - into the number of turns greater that 270 degrees, then back to a 2d

        - into the number of turns greater that 270 degrees, then back to a 2d position vector?
    - is this naturally what auto encoders do?
    - what is the difference between a dimension and a feature?
- what is this like? electro magnetic forces acting on particles? gravity acting on particles, a network of forces? ... what techinques have they developed?
- data visualisation technique researchers dont seem to know exactally what they are looking to visualise. we need to rigorously define what it is we are trying to do
- how does this work for unsupervised data?
    - supervised data can ... nice visualisations?
    - unsupervised? we are clustering based on similarities? we can look at how it has group datapoints and see the patterns?

Resources and references

- t-SNE on SciKit learn
- LAURENS VAN DER MAATEN
-

Glossary

- Local structure: the similarities/differences of
- Global structure: the