

Valamelyik tétel

Nagy Dániel

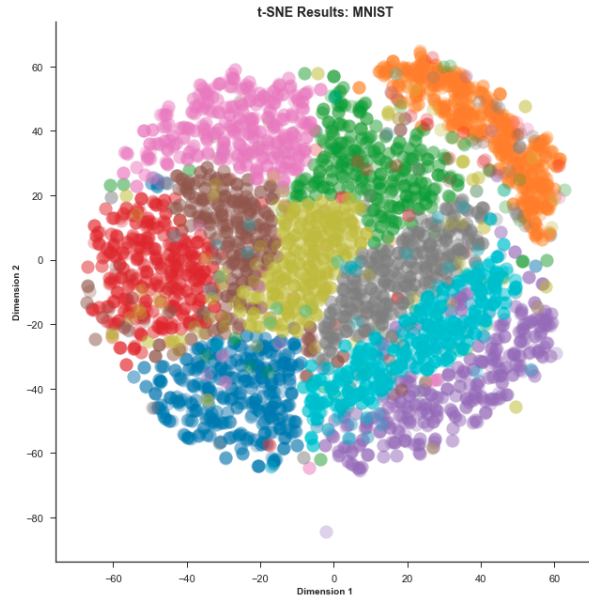
2019. június 5.

1. Fejezet címe

Random tétel

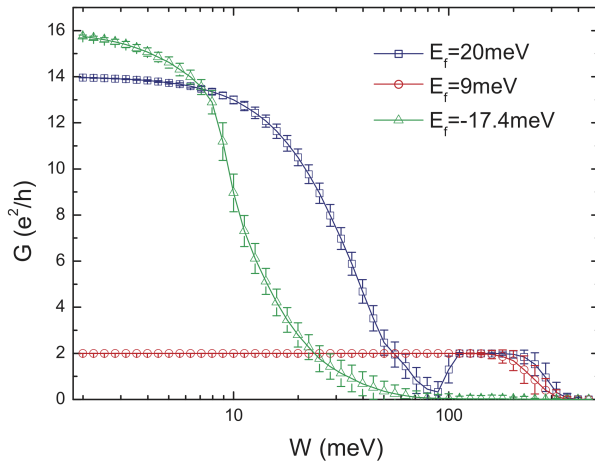
1.1. Hasznos dolgok

- Minden képet, egyéb anyagot rakjunk a <tétel száma>/media mappába!
- Egy szimpla kép [1](#).

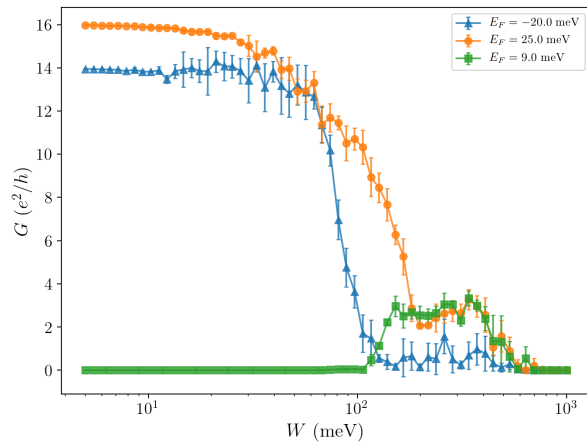


1. ábra. t-SNE plot for MNIST dataset [1]

- Két kép egymás mellett:



2. ábra. Első kép



3. ábra. Második kép

- Ez lesz egy url (hosszú url is jól működik): <https://towardsdatascience.com/time-series-analysis-and-climate-change-7bb4371021e>.
- Egy egyszerű képlet számozva:

$$(i\hbar\gamma^\mu\partial_\mu - mc)|\psi\rangle = 0 \quad (1)$$

Így hivatkozom: Dirac-egyenlet (1).

- Egy számozatlan képlet:

$$\hat{F}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt e^{-i\omega t} f(t)$$

- bra-ket jelölés: $|a\rangle \langle b| \langle a|b|c\rangle \left\langle \psi \left| \sum_{j=1}^{\infty} c_j \right| \psi \right\rangle$
- Deriválásnál, integrálásnál használjuk az egyenes d (\code{dd}) parancsot:

$$\int d\omega \frac{dy}{dx}$$

- Több összefüggés egymás alatt: `\begin{align*} ... \end{align*}`

$$\begin{aligned} [b_l, b_m^\dagger] &= \delta_{lm} \\ [b_l^\dagger, b_m^\dagger] &= [b_l, b_m] = 0, \end{aligned}$$

- Hosszú, többsoros képlet:

$$\begin{aligned} A &= B + C + D \\ &= \int dx f_B(x) + \int dx f_C(x) + \int dx f_D(x) \\ &= 1 - \frac{1}{\sqrt{2\pi}} \int d\omega Y(\omega) \end{aligned}$$

- Ha kódot szeretnénk képpé alakítani, itt egy hasznos tool [https://carbon.now.sh/?bg=rgba\(171%2C184%2C195%2C0\)&t=monokai&wt=none&l=python&ds=false&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=8px&ph=5px&ln=false&fm=Source%20Code%20Pro&fs=14px&lh=133%25&si=false&es=2x&wm=false](https://carbon.now.sh/?bg=rgba(171%2C184%2C195%2C0)&t=monokai&wt=none&l=python&ds=false&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=8px&ph=5px&ln=false&fm=Source%20Code%20Pro&fs=14px&lh=133%25&si=false&es=2x&wm=false) Eredmény:
- Itt egy fizwebes referencia [2].
- Itt meg a KNN wikipédia referencia [3].

```

hamiltonian = """
(C-D*(k_x**2+k_y**2))*identity(4)
+ (M-B*(k_x**2+k_y**2))*kron(sigma_0, sigma_z)
+ A*k_x*kron(sigma_z, sigma_x)
+ A*k_y*kron(sigma_z, sigma_y)
+ V(x,y)*identity(4)
"""

template = kwant.continuum.discretize(hamiltonian, grid=a)

def shape(site):
    (x, y) = site.pos
    return (0 <= y < W and 0 <= x < L)

syst = kwant.Builder()
syst.fill(template, shape, (0, 0))

```

4. ábra. Kép forrása: <https://carbon.now.sh/>

1.2. Blabla

Hivatkozások

- [1] Article about t-sne on TowardsDatascience <https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1>.
- [2] Fizweben elérhető jegyzet: http://fizweb.elte.hu/download/Fizikus-MSc/Reszecskefizika/partphys_notes_2018.pdf.
- [3] K-nearest neighbors https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.