2020 Global Smart Industry Conference (Chelyabinsk, Russian Federation, November 17-19, 2020)

N-dimensional model for visual representation of linear programming problem

(N-мерная модель визуального представления задачи линейного программирования)

N.A. Olkhovsky

South Ural State University
(national research university)

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00092-а и Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

Linear programming problem

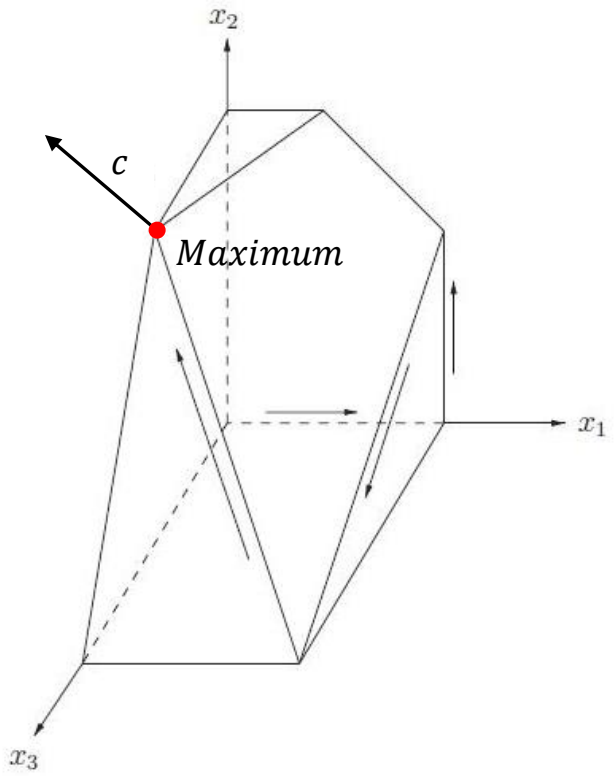
$Ax \leq b$ $\qquad$ $F(x) = \langle c, x \rangle$

$A \in \mathbb{R}^{m \times n}$ $\qquad$ $c \in \mathbb{R}^n$

$x \in \mathbb{R}^n$

$b \in \mathbb{R}^m$

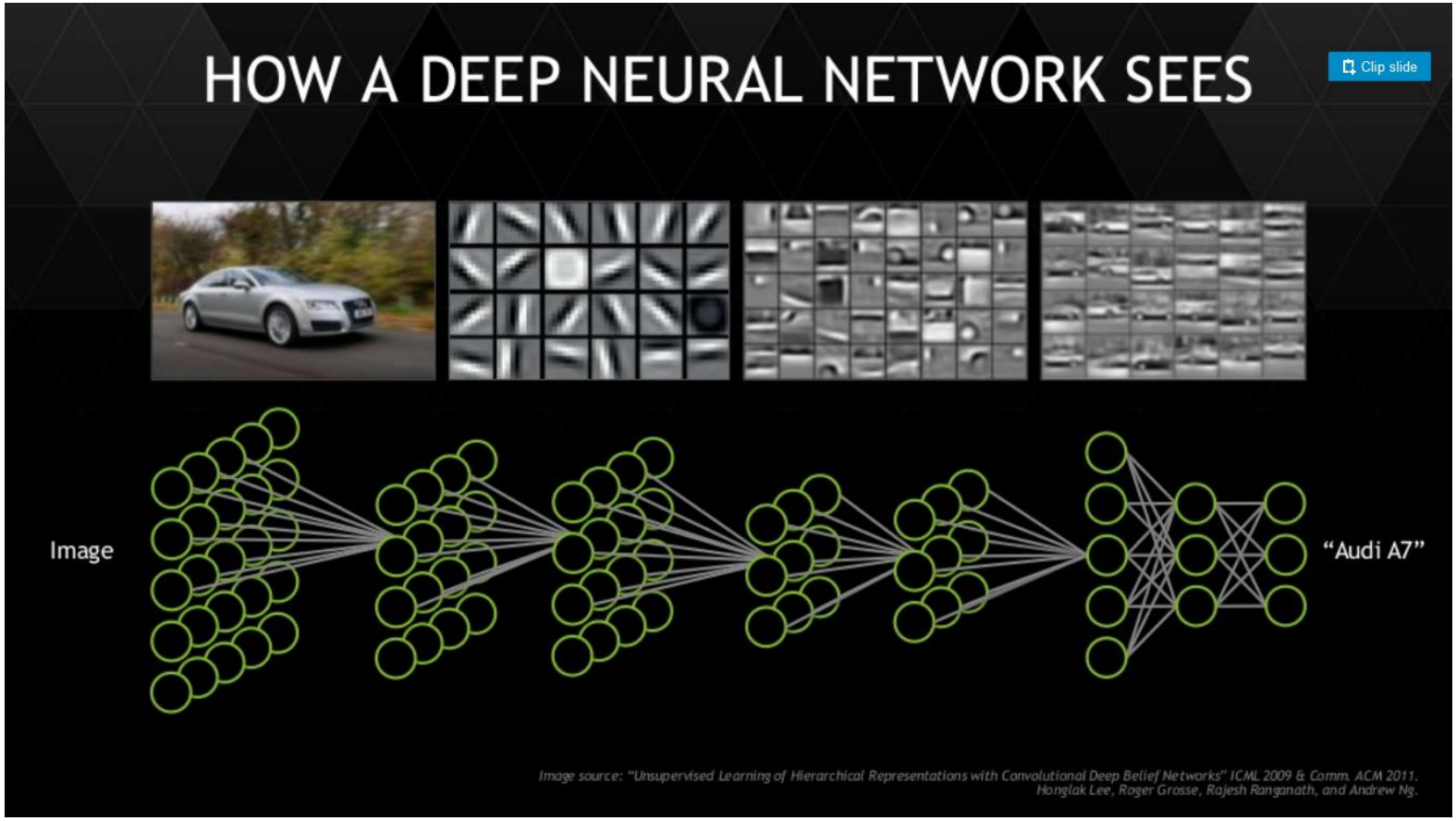
$$\hat{x} = argmax\{F(x) \mid Ax \leq b\}$$



Problem

To investigate the possibility of using artificial neural networks to solve the LP problem

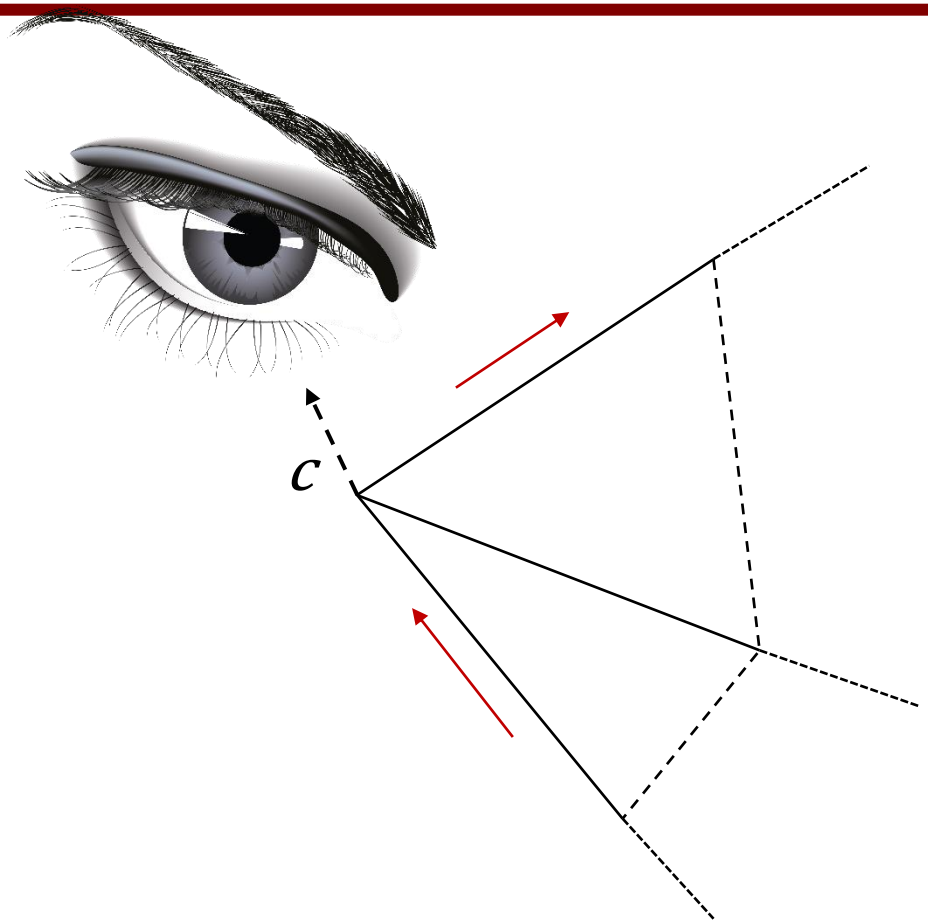
Image analysis with ANN



Idea

To visualize linear programming problem in N-dimensional space

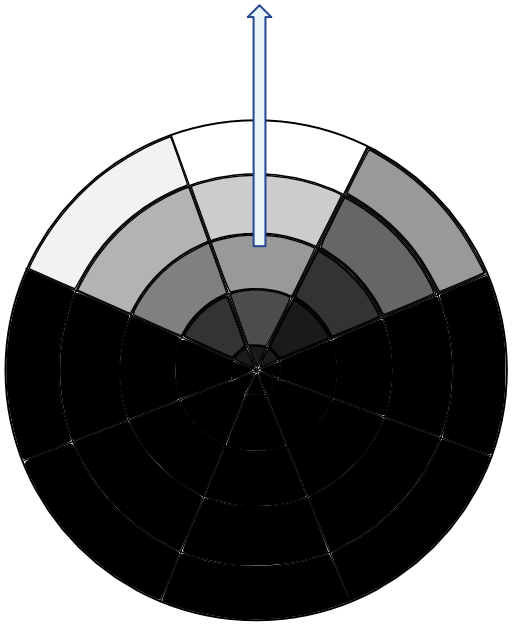
Vertex of polyhedron

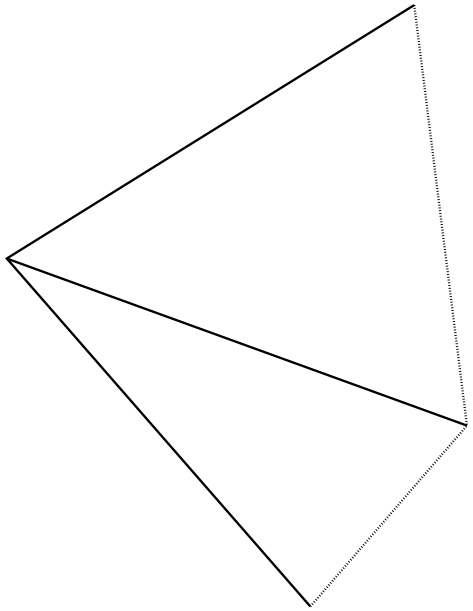


What if we create a virtual retina able to look at every single vertex in N-dimensional polyhedron and find the optimal edge to go.

Grayscale image of the vertex

Best way

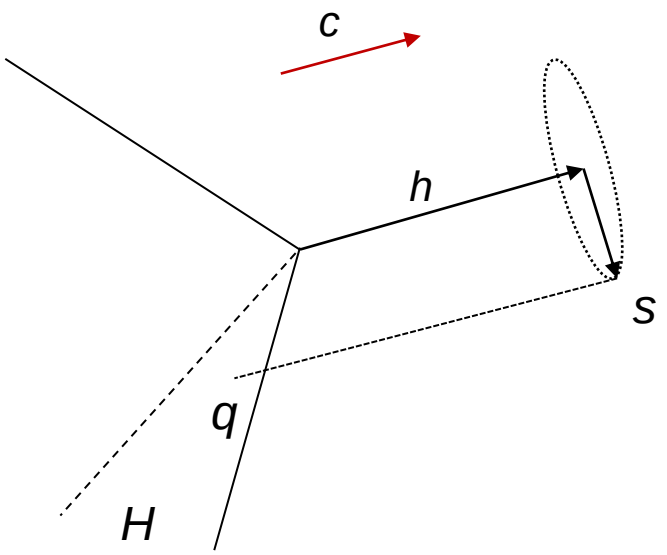




Ray tracing

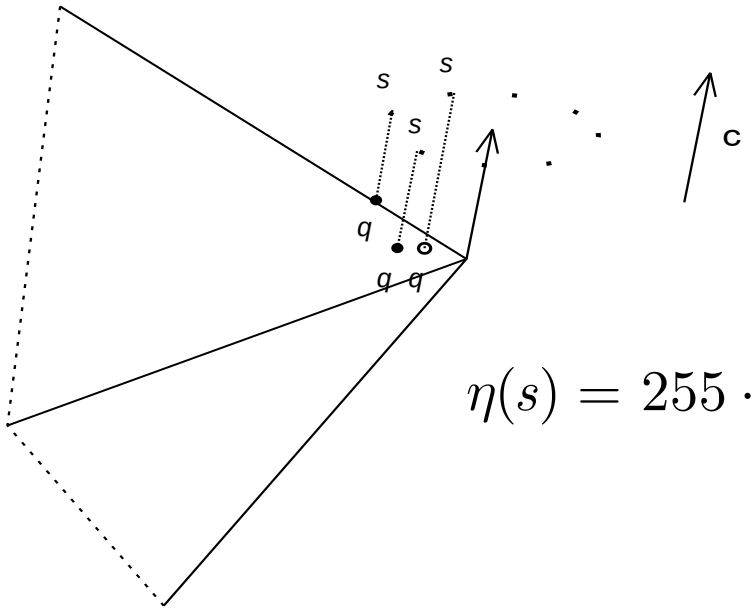
$$H \leftrightarrow \left\langle a, x \right\rangle = b$$

$$q - s = h \frac{b - \left\langle a, g \right\rangle}{\left\langle a, c \right\rangle}$$



$$h = \lambda \frac{c}{\|c\|}$$

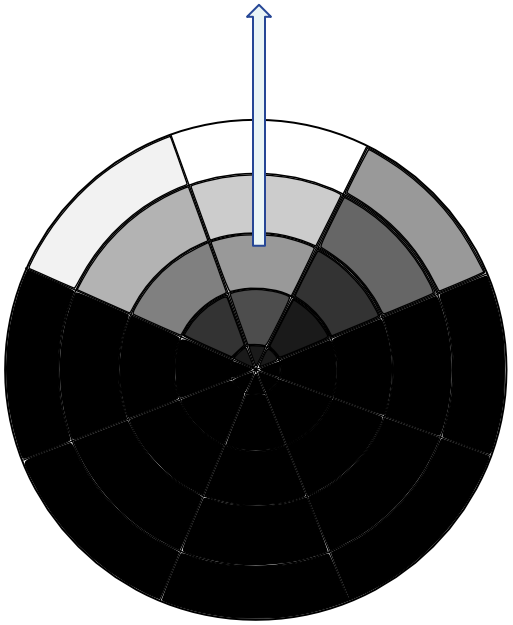
Gray scaling



$$\eta(s) = 255 \cdot \frac{w - \left|q - s\right|}{w - v}$$

$$w = \max \left|q - s\right|$$
$$v = \min \left|q - s\right|$$

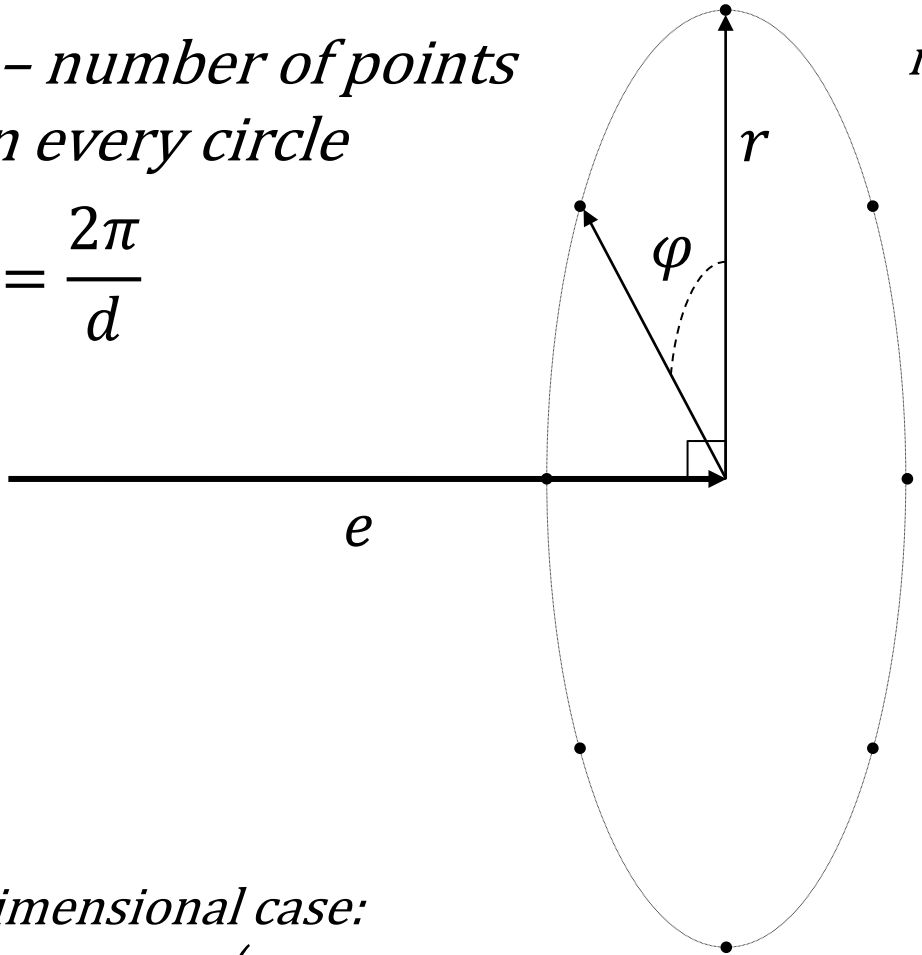
Best way



Define virtual retina

d – number of points on every circle

$$\varphi = \frac{2\pi}{d}$$



n-dimensional sphere of points:

$$S_r^{(e)} = (r, \varphi_1, \dots, \varphi_{n-2}, \theta)$$

$$\varphi_1 = \frac{\pi}{2}$$

$$\varphi_{2\dots n-2} = \frac{2k\pi}{d}, k = 0,1, \dots, \frac{d}{2}-1$$

$$\theta = \frac{2k\pi}{d}, k = 0,1, \dots, d-1$$

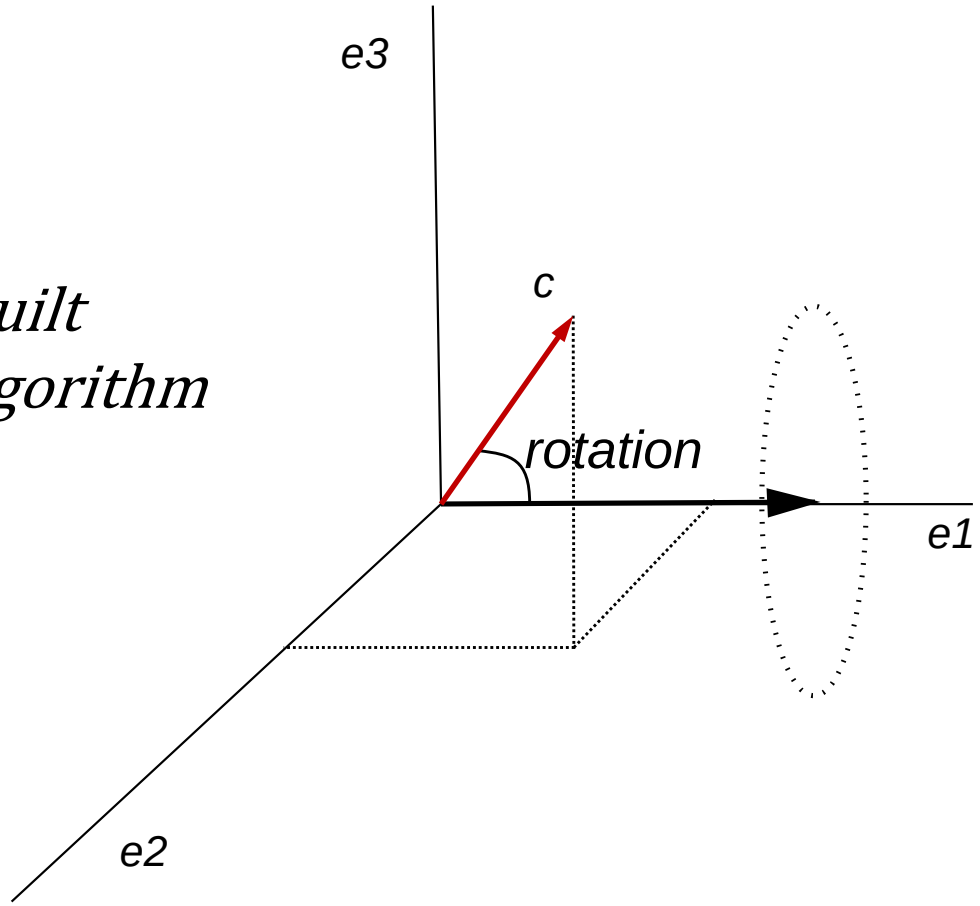
3-dimensional case:

$$\nabla\varphi = \frac{\pi}{2}\left(\nabla\theta \in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\right\}\left(make\ S_r^{(e)} = (r, \varphi_1, \theta)\right)\right)$$

Rotation to the target vector

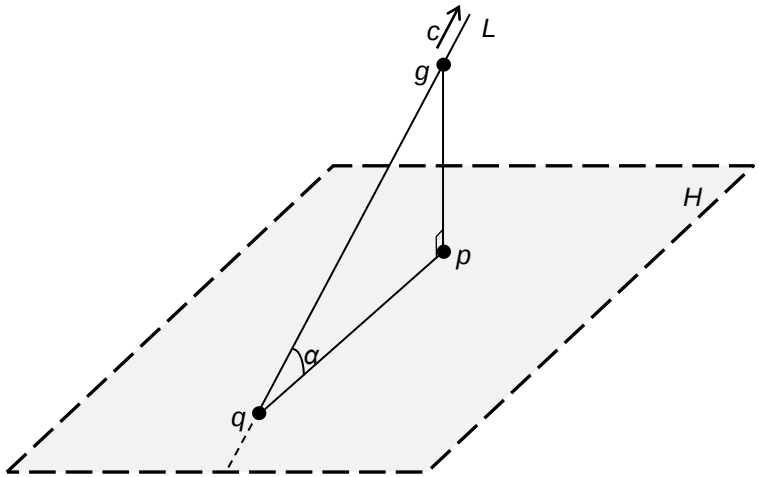
$$s' = s \cdot M$$

M - rotation matrix built by Aguilera-Perez Algorithm



Thank you!

Projection to polyhedron



$$\boldsymbol{q = g + c\frac{b - \left\langle a, g \right\rangle}{\left\langle a, c \right\rangle}}$$

$g, c, a \in \mathbb{R}^n,\ b \in \mathbb{R}$

$$\left\langle a, x \right\rangle = b$$

$$L = g + ct$$

$$q = g + ct_0$$

$$\left\langle a, g + ct_0 \right\rangle = b$$

$$t_0 = \frac{b - \left\langle a, g \right\rangle}{\left\langle a, c \right\rangle}$$

References

1. Aguilera A., Pérez-Aguila R. General n-dimensional rotations // Proc. WSCG SHORT Commun. Pap. 2004. P. 1–8.

2. Blumenson L.. E.. A Derivation of n-Dimensional Spherical Coordinates // Am. Math. Mon. 1960. Vol. 67, no. 1. P. 63–66.