Neurodynamic Optimization: New Models and kWTA Applications

Jun Wang
jwang@mae.cuhk.edu.hk

Department of Mechanical & Automation Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
http://www.mae.cuhk.edu.hk/~jwang
Introduction

Optimization is ubiquitous in nature and society.
Introduction

Optimization is ubiquitous in nature and society.

Optimization arises in a wide variety of scientific problems.
Introduction

Optimization is ubiquitous in nature and society.

Optimization arises in a wide variety of scientific problems.

Optimization is an important tool for design, planning, control, operation, and management of engineering systems.
Problem Formulation

Consider a general optimization problem:

\[ \text{OP}_1 : \text{Minimize} \quad f(x) \]
\[ \text{subject to} \quad c(x) \leq 0, \]
\[ d(x) = 0, \]

where \( x \in \mathbb{R}^n \) is the vector of decision variables, \( f(x) \) is an objective function, \( c(x) = [c_1(x), \ldots, c_m(x)]^T \) is a vector-valued function, and \( d(x) = [d_1(x), \ldots, d_p(x)]^T \) a vector-valued function.
Problem Formulation

Consider a general optimization problem:

$$\text{OP}_1 : \text{Minimize } f(x)$$
$$\text{subject to } c(x) \leq 0, \quad d(x) = 0,$$

where $x \in \mathbb{R}^n$ is the vector of decision variables, $f(x)$ is an objective function, $c(x) = [c_1(x), \ldots, c_m(x)]^T$ is a vector-valued function, and $d(x) = [d_1(x), \ldots, d_p(x)]^T$ a vector-valued function.

If $f(x)$ and $c(x)$ are convex and $d(x)$ is affine, then OP is a convex programming problem CP. Otherwise, it is a nonconvex program.
Quadratic and Linear Programs

\[ \text{QP}_1 : \text{minimize} \quad \frac{1}{2} x^T Q x + q^T x \]
subject to \[ A x = b, \]
\[ l \leq C x \leq h, \]

where \( Q \in \mathbb{R}^{n \times n} \), \( q \in \mathbb{R}^n \), \( A \in \mathbb{R}^{m \times n} \), \( b \in \mathbb{R}^m \), \( C \in \mathbb{R}^{n \times n} \), \( l \in \mathbb{R}^n \), \( h \in \mathbb{R}^n \).
Quadratic and Linear Programs

\[ QP_1 : \text{minimize} \quad \frac{1}{2} x^T Q x + q^T x \]
subject to \[ Ax = b, \quad l \leq C x \leq h, \]
where \( Q \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \]
\( b \in \mathbb{R}^m, C \in \mathbb{R}^{n \times n}, l \in \mathbb{R}^n, h \in \mathbb{R}^n. \)

When \( Q = 0, \) and \( C = I, \) \( QP_1 \) becomes a linear program with equality and bound constraints:

\[ LP_1 : \text{minimize} \quad q^T x \]
subject to \[ Ax = b, \quad l \leq x \leq h. \]
Dynamic Optimization

In many applications (e.g., online pattern recognition and onboard signal processing), real-time solutions to optimization problems are necessary or desirable.
Dynamic Optimization

In many applications (e.g., online pattern recognition and onboard signal processing), real-time solutions to optimization problems are necessary or desirable.

For such applications, classical optimization techniques may not be competent due to the problem dimensionality and stringent requirement on computational time.
Dynamic Optimization

In many applications (e.g., online pattern recognition and onboard signal processing), real-time solutions to optimization problems are necessary or desirable.

For such applications, classical optimization techniques may not be competent due to the problem dimensionality and stringent requirement on computational time.

It is computationally challenging when optimization procedures have to be performed in real time to optimize the performance of dynamical systems.
Dynamic Optimization

In many applications (e.g., online pattern recognition and onboard signal processing), real-time solutions to optimization problems are necessary or desirable.

For such applications, classical optimization techniques may not be competent due to the problem dimensionality and stringent requirement on computational time.

It is computationally challenging when optimization procedures have to be performed in real time to optimize the performance of dynamical systems.

One very promising approach to dynamic optimization is to apply artificial neural networks.
Neurodynamic Optimization

Because of the inherent nature of parallel and distributed information processing in neural networks, the convergence rate of the solution process is not decreasing as the size of the problem increases.
Neurodynamic Optimization

Because of the inherent nature of parallel and distributed information processing in neural networks, the convergence rate of the solution process is not decreasing as the size of the problem increases.

Neural networks can be implemented physically in designated hardware such as ASICs where optimization is carried out in a truly parallel and distributed manner.
Neurodynamic Optimization

Because of the inherent nature of parallel and distributed information processing in neural networks, the convergence rate of the solution process is not decreasing as the size of the problem increases.

Neural networks can be implemented physically in designated hardware such as ASICs where optimization is carried out in a truly parallel and distributed manner.

This feature is particularly desirable for dynamic optimization in decentralized decision-making situations.
Existing Approaches

In their seminal work, Tank and Hopfield (1985, 1986) applied the Hopfield networks for solving a linear program and the traveling salesman problem.
Existing Approaches

In their seminal work, Tank and Hopfield (1985, 1986) applied the Hopfield networks for solving a linear program and the traveling salesman problem.

Kennedy and Chua (1988) developed a neural network for nonlinear programming, which contains finite penalty parameters and thus its equilibrium points correspond to approximate optimal solutions only.
Existing Approaches

In their seminal work, Tank and Hopfield (1985, 1986) applied the Hopfield networks for solving a linear program and the traveling salesman problem.

Kennedy and Chua (1988) developed a neural network for nonlinear programming, which contains finite penalty parameters and thus its equilibrium points correspond to approximate optimal solutions only.

Existing Approaches

In their seminal work, Tank and Hopfield (1985, 1986) applied the Hopfield networks for solving a linear program and the traveling salesman problem.

Kennedy and Chua (1988) developed a neural network for nonlinear programming, which contains finite penalty parameters and thus its equilibrium points correspond to approximate optimal solutions only.


Existing Approaches (cont’d)

A recurrent neural network for quadratic optimization with bounded variables only by Bouzerdoum and Pattison (1993).
Existing Approaches (cont’d)

A recurrent neural network for quadratic optimization with bounded variables only by Bouzerdoum and Pattison (1993).

Existing Approaches (cont’d)

A recurrent neural network for quadratic optimization with bounded variables only by Bouzerdoum and Pattison (1993).


The primal-dual networks for linear and quadratic programming by Xia (1996, 1997).
Existing Approaches (cont’d)

A recurrent neural network for quadratic optimization with bounded variables only by Bouzerdoum and Pattison (1993).


The primal-dual networks for linear and quadratic programming by Xia (1996, 1997).

Existing Approaches (cont’d)

Existing Approaches (cont’d)


A two-layer network for convex programming subject to nonlinear inequality constraints by Xia and Wang (2004).
Existing Approaches (cont’d)


A two-layer network for convex programming subject to nonlinear inequality constraints by Xia and Wang (2004).

Existing Approaches (cont’d)


A two-layer network for convex programming subject to nonlinear inequality constraints by Xia and Wang (2004).


Primal-Dual Network

The primal-dual network for solving LP$^a$:

\[
\begin{align*}
\epsilon \frac{dx}{dt} &= -(q^T x - b^T y)q - A^T (Ax - b) + x^+, \\
\epsilon \frac{dy}{dt} &= (q^T x - b^T y)b,
\end{align*}
\]

where $\epsilon > 0$ is a scaling parameter, $x \in \mathbb{R}^n$ is the primal state vector, $y \in \mathbb{R}^m$ is the dual (hidden) state vector, $x^+ = (x_1^+, \ldots, x_n^+)^T$, and $x_i^+ = \max\{0, x_i\}$.

---

$^a$Y. Xia, “A new neural network for solving linear and quadratic programming problems,”

*IEEE Transactions on Neural Networks*, vol. 7, no. 6, 1544-1548, 1996.
Primal-Dual Network

The primal-dual network for solving $\text{LP}_2$:

$$
\varepsilon \frac{dx}{dt} = -(q^T x - b^T y)q - A^T(Ax - b) + x^+,
$$

$$
\frac{dy}{dt} = (q^T x - b^T y)b,
$$

where $\varepsilon > 0$ is a scaling parameter, $x \in \mathbb{R}^n$ is the primal state vector, $y \in \mathbb{R}^m$ is the dual (hidden) state vector, $x^+ = (x_1^+, \ldots, x_n^+)^T$, and $x_i^+ = \max\{0, x_i\}$. The network is globally convergent to an optimal solution to $\text{LP}_1$.

---

\textsuperscript{1}Y. Xia, “A new neural network for solving linear and quadratic programming problems,” 

*IEEE Transactions on Neural Networks*, vol. 7, no. 6, 1544-1548, 1996.
Lagrangian Network for QP

If $C = 0$ in $QP_1$:

$$\epsilon \frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -Qx(t) - A^T y(t) - q, \\ Ax - b \end{pmatrix}.$$ 

where $\epsilon > 0$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$.

It is globally exponentially convergent to the optimal solution.$^a$

---

Projection Network

A recurrent neural network called the projection network was developed for optimization with bound constraints only\(^a\)

\[
\epsilon \frac{dx}{dt} = -x + g(x - \nabla f(x)),
\]

where \(g(\cdot)\) is a vector-valued piecewise-linear activation function.

Piecewise-Linear Activation Function

\[ g(x_i) = \begin{cases} 
  l_i & x_i < l_i \\
  x_i & l_i \leq x_i \leq h_i \\
  h_i & x_i > h_i.
\end{cases} \]
Two-layer Projection Network for QP₁

If \( C = I \) in QP₁, let \( \alpha = 1 \) in the two-layer neural network for CP:

\[
\epsilon \frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x + g((I - Q)x + A^T y - q) \\ -Ax + b \end{pmatrix}.
\]

where \( \epsilon > 0, x \in \mathbb{R}^n, y \in \mathbb{R}^m, \)
\( g(x) = [g(x_1), \ldots, g(x_n)]^T \) is the piecewise-linear activation function defined before.

It is globally asymptotically convergent to the optimal solution\(^d\).

General projection Network for QP_1

The dynamic equation of the general projection neural network (GPNN):

\[
\epsilon \frac{dz}{dt} = (M + N)^T \{-Nz + g((N - M)z)\},
\]

where \( \epsilon > 0 \) and \( z = (x^T, y^T)^T \) is the state vector,

\[
M = \begin{pmatrix} Q & -A^T \\ 0 & I \end{pmatrix}, \quad N = \begin{pmatrix} I & 0 \\ A & 0 \end{pmatrix}.
\]

The GPNN is globally convergent to an exact solution of the problem\(^a\).

Dual Network for QP$_2$

For strictly convex QP$_2$, $Q$ is invertible. The dynamic equation of the dual network:

$$\epsilon \frac{dy(t)}{dt} = -CQ^{-1}C^T y + g \left( CQ^{-1}C^T y - y - Cq \right) + Cq + b,$$

$$x(t) = Q^{-1}C^T y - q,$$

where $\epsilon > 0$. It is also globally exponentially convergent to the optimal solution.$^a$$^b$

---


Simplified Dual Net for QP

For strictly convex QP, $Q$ is invertible. The dynamic equation of the simplified dual network is:

$$
\epsilon \frac{du}{dt} = -Cx + g(Cx - u),
$$

$$
x = Q^{-1}(A^T y + C^T u - q),
$$

$$
y = (AQ^{-1}A^T)^{-1} [-AQ^{-1}C^T u + AQ^{-1}q + b],
$$

where $u \in \mathbb{R}^n$ is the state vector, $\epsilon > 0$. It is proven to be globally asymptotically convergent to the optimal solution.

---

Illustrative Example

minimize \[3x_1^2 + 3x_2^2 + 4x_3^2 + 5x_4^2 + 3x_1 x_2 + 5x_1 x_3 + x_2 x_4 - 11x_1 - 5x_4\]

subject to \[3x_1 - 3x_2 - 2x_3 + x_4 = 0,\]
\[4x_1 + x_2 - x_3 - 2x_4 = 0,\]
\[-x_1 + x_2 \leq -1,\]
\[-2 \leq 3x_1 + x_3 \leq 4.\]

\[Q = \begin{bmatrix} 6 & 3 & 5 & 0 \\ 3 & 6 & 0 & 1 \\ 5 & 0 & 8 & 0 \\ 0 & 1 & 0 & 10 \end{bmatrix}, \quad q = \begin{bmatrix} -11 \\ 0 \\ 0 \\ -5 \end{bmatrix},\]
\[A = \begin{bmatrix} 3 & -3 & -2 & 1 \\ 4 & 1 & -1 & -2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\]
\[C = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \end{bmatrix}, \quad l = \begin{bmatrix} -\infty \\ -2 \end{bmatrix}, \quad h = \begin{bmatrix} -1 \\ 4 \end{bmatrix}.\]
Illustrative Example (cont’d)

The simplified dual neural network for solving this quadratic program needs only two neurons only.

In contrast, the Lagrange neural network needs twelve neurons.

The primal-dual neural network needs nine neurons.

The dual neural network needs four neurons.
Illustrative Example (cont’d)

Transient behaviors of the state vector $u$. 

Computational Intelligence Laboratory, CUHK – p. 20/69
Illustrative Example (cont’d)

Transient behaviors of the output vector $x$. 

Computational Intelligence Laboratory, CUHK – p. 21/69
Trajectories of $x_1$ and $x_2$ from different initial states.
Illustrative Example (cont’d)

Trajectories of $x_3$ and $x_4$ from different initial states.
Improved Dual Net for special QP_1

For special convex QP_1 when \( Q = I \), the dynamic equation of the improved dual network^a:

\[
\begin{align*}
\frac{\epsilon dy}{dt} &= -y + (y + Ax - b)^+, \\
\frac{\epsilon dz}{dt} &= -Cx + d,
\end{align*}
\]

\[
x = g_\Omega(-A^T y + C^T z - p),
\]

where \( g(\cdot) \) and \((\cdot)^+\) are two activation functions. It is proven to be globally convergent to the optimal solution.

A One-layer Net for LP

A new recurrent neural network model with a discontinuous activation function was recently developed for linear programming LP\(^a\):

\[
\epsilon \frac{dx}{dt} = -Px - \sigma(I - P)g(x) + s,
\]

where \(g(x) = (g_1(x_1), g_2(x_2), \ldots, g_n(x_n))^T\) is the vector-valued activation function, \(\epsilon\) is a positive scaling constant, \(\sigma\) is a nonnegative gain parameter, \(P = A^T(AA^T)^{-1}A\), and

\[
s = -(I - P)q + A^T(AA^T)^{-1}b.
\]

Activation Function

A discontinuous activation function is defined as follows: For $i = 1, 2, \ldots, n$;

$$g_i(x_i) = \begin{cases} 
1, & \text{if } x_i > h_i, \\
[0, 1], & \text{if } x_i = h_i, \\
0, & \text{if } x_i \in (l_i, h_i), \\
[-1, 0], & \text{if } x_i = l_i, \\
-1, & \text{if } x_i < l_i.
\end{cases}$$
Activation Function (cont’d)

\[ g_i(x_i) \]

\[ l_i \]

\[ 0 \]

\[ h_i \]

\[ x_i \]

\[ 1 \]

\[ -1 \]
Convergence Results

The neural network is globally convergent to an optimal solution of LP_1 with \( C = I \), if \( \bar{\Omega} \subset \Omega \), where \( \bar{\Omega} \) is the equilibrium point set and \( \Omega = \{ x | l \leq x \leq h \} \).
Convergence Results

The neural network is globally convergent to an optimal solution of LP$_1$ with $C = I$, if $\bar{\Omega} \subset \Omega$, where $\bar{\Omega}$ is the equilibrium point set and $\Omega = \{x | l \leq x \leq h\}$.

The neural network is globally convergent to an optimal solution of LP$_1$ with $C = I$, if it has a unique equilibrium point and $\sigma \geq 0$ when $(I - P)c = 0$ or one of the following conditions holds when $(I - P)c \neq 0$:

(i) $\sigma \geq \|(I - P)c\|_p/ \min_{\gamma \in X}^{+} \|(I - P)\gamma\|_p$ for $p = 1, 2, \infty$, or

(ii) $\sigma \geq c^T (I - P)c/ \min_{\gamma \in X}^{+} \{c^T (I - P)\gamma\}$,

where $X = \{-1, 0, 1\}^n$. 

Computational Intelligence Laboratory, CUHK – p. 28/69
Simulation Results

Consider the following LP problem:

minimize \quad 4x_1 + x_2 + 2x_3,
subject to \quad x_1 - 2x_2 + x_3 = 2,
\quad -x_1 + 2x_2 + x_3 = 1,
\quad -5 \leq x_1, x_2, x_3 \leq 5.

According to the above condition, the lower bound of \( \sigma \) is 9.
Transient behaviors of the states with $\sigma = 15$. 

Transient behaviors of the states with $\sigma = 9$. 
Simulation Results (cont’d)

Transient behaviors of the states with $\sigma = 5$. 

Computational Intelligence Laboratory, CUHK – p. 32/69
Transient behaviors of the states with $\sigma = 3$. 
A New One-layer Net for QP

A new one-layer recurrent neural net was recently developed:\(^a\):

\[
\epsilon \frac{dz}{dt} = -(I - P)z - [(I - P)Q + \alpha P]g(z) + q,
\]

\[
x = ((I - P)Q + \alpha P)^{-1}(-(I - P)z + s),
\]

where \(\epsilon\) is a positive scaling constant, \(\alpha > 0\) is a parameter, \(s = -q + Pq + \alpha A^T(AA^T)^{-1}b\), and \(g(\cdot)\) is a vector-valued activation function.

Activation Function

The following hard-limiting activation function is defined:

\[
g_i(z_i) \begin{cases} 
= h_i, & \text{if } z_i > 0, \\
\in [l_i, h_i], & \text{if } z_i = 0, \\
= l_i, & \text{if } z_i < 0.
\end{cases}
\]

If \( l_i \neq h_i \), then \( g_i \) is discontinuous.

When \( z_i = 0 \), \( g_i(z_i) \) can take any values between \( l_i \) and \( h_i \).
Activation Function (cont’d)

\[ g_i(z_i) \]

- \( h_i \)
- \( l_i \)
- \( 0 \)
- \( z_i \)
Convergence results

Assume that $Q$ is positive definite. If $\alpha \geq \frac{\lambda_{\text{max}}(Q)}{2}$ or $\alpha \geq \frac{\text{trace}(Q)}{2}$, then the state vector $z(t)$ of the neural network is globally convergent to an equilibrium point and the output vector $x(t)$ is globally convergent to an optimal solution of QP.

Assume that the objective function $f(x)$ is strictly convex on the set $S = \{x \in \mathbb{R}^n : Ax = b\}$. If

$$\alpha > \frac{\lambda_{\text{max}}(Q^2)\lambda_{\text{max}}(Q^{-1})}{4},$$

then the state vector $z(t)$ of the neural network is globally convergent to an equilibrium point and the output vector $x(t)$ is globally convergent to an optimal solution of QP.
Illustrative Example

Consider the following QP problem:

minimize \( f(x) = -0.5x_1^2 + x_2^2 + 2x_1x_2 + 6x_1 - 2 \)
subject to \( 3x_1 - 2x_2 = 1, \)
\( 0 \leq x_1, x_2 \leq 10. \)

As \( Q = \begin{pmatrix} -0.5 & 1 \\ 1 & 1 \end{pmatrix} \)

is not positive definite, the objective function is not convex everywhere. However, if we substitute \( x_1 = 2x_2/3 + 1/3 \) into the objective function, then \( \hat{f}(x_2) = 19x_2^2/9 + 22x_2/9 - 35/18 \) is convex.
Illustrative Example (cont’d)

The state variables of the new network.
Illustrative Example (cont’d)

The output variables of the new network.
Illustrative Example (cont’d)

Phase plot of the output variables.
Illustrative Example (cont’d)

The simulation result of the dual network.

\begin{align*}
\text{sign}(x_1) \log_{10}(|x_1|) \\
\text{sign}(x_2) \log_{10}(|x_2|)
\end{align*}
Illustrative Example (cont’d)

The simulation result of the projection network.
## Model Comparisons for QP₁

<table>
<thead>
<tr>
<th>model</th>
<th>layers</th>
<th>neurons</th>
<th>connections</th>
<th>convergence condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lagrangian network</td>
<td>2</td>
<td>$3n + m$</td>
<td>$n^2 + 2mn$</td>
<td>$f(x)$ is strictly convex</td>
</tr>
<tr>
<td>Primal-dual network</td>
<td>2</td>
<td>$n + m$</td>
<td>$3n^2 + 3mn$</td>
<td>$f(x)$ is convex</td>
</tr>
<tr>
<td>General projection net</td>
<td>2</td>
<td>$n + m$</td>
<td>$n^2 + 2mn$</td>
<td>$f(x)$ is convex</td>
</tr>
<tr>
<td>Dual network</td>
<td>1</td>
<td>$n + m$</td>
<td>$(n + m)^2$</td>
<td>$f(x)$ is strictly convex</td>
</tr>
<tr>
<td>Simplified dual network</td>
<td>1</td>
<td>$n$</td>
<td>$n^2$</td>
<td>$f(x)$ is strictly convex</td>
</tr>
<tr>
<td>New neural network</td>
<td>1</td>
<td>$n$</td>
<td>$2n^2$</td>
<td>$f(x)$ is strictly convex on $S$</td>
</tr>
</tbody>
</table>

where $S = \{ x \in \mathbb{R}^n : Ax = b \}$. 

---

*Computational Intelligence Laboratory, CUHK – p. 44*
$k$ Winners Take All Operation

The $k$-winners-take-all ($k$WTA) operation is to select the $k$ largest inputs out of $n$ inputs ($1 \leq k \leq n$).
\( k \) Winners Take All Operation

The \( k \)-winners-take-all (\( k \)WTA) operation is to select the \( k \) largest inputs out of \( n \) inputs (\( 1 \leq k \leq n \)).

The \( k \)WTA operation has important applications in machine learning, such as \( k \)-neighborhood classification, \( k \)-means clustering, etc.
\( k \) Winners Take All Operation

The \( k \)-winners-take-all (\( k \)WTA) operation is to select the \( k \) largest inputs out of \( n \) inputs (\( 1 \leq k \leq n \)).

The \( k \)WTA operation has important applications in machine learning, such as \( k \)-neighborhood classification, \( k \)-means clustering, etc.

As the number of inputs increases and/or the selection process should be operated in real time, parallel algorithms and hardware implementation are desirable.
$k$WTA Problem Formulations

The $k$WTA function can be defined as:

$$x_i = f(u_i) = \begin{cases} 
1, & \text{if } u_i \in \{k \text{ largest elements of } u\}, \\
0, & \text{otherwise},
\end{cases}$$

where $u \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$ is the input vector and output vector, respectively.
The $k$WTA function can be defined as:

\[
x_i = f(u_i) = \begin{cases} 
1, & \text{if } u_i \in \{k \text{ largest elements of } u\}, \\
0, & \text{otherwise},
\end{cases}
\]

where $u \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$ is the input vector and output vector, respectively. The $k$WTA solution can be determined by solving the following linear integer program:

\[
\begin{align*}
\text{minimize} & \quad -\sum_{i=1}^{n} u_i x_i, \\
\text{subject to} & \quad \sum_{i=1}^{n} x_i = k, \\
& \quad x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n.
\end{align*}
\]
$k$WTA Problem Formulations

If the $k$th and $(k + 1)$th largest elements of $u$ are different (denoted as $\bar{u}_k$ and $\bar{u}_{k+1}$ respectively), the $k$WTA problem is equivalent to the following LP or QP problems:

\[
\begin{align*}
\text{minimize} & \quad -u^T x \quad \text{or} \quad \frac{a}{2} x^T x - u^T x, \\
\text{subject to} & \quad \sum_{i=1}^{n} x_i = k, \\
& \quad 0 \leq x_i \leq 1, \quad i = 1, 2, \ldots, n,
\end{align*}
\]

where $a \leq \bar{u}_k - \bar{u}_{k+1}$ is a positive constant.
QP-based Primal-Dual Network

The primal-dual network based on the QP formulation needs $3n + 1$ neurons and $6n + 2$ connections, and its dynamic equations can be written as:

\[
\begin{align*}
\epsilon \frac{dx}{dt} &= -(1 + a)(x - (x + ve + w - ax + u)^+ - (e^T x - k)e - x - y + e) \\
\epsilon \frac{dy}{dt} &= -y + (y + w)^+ - x - y + e \\
\epsilon \frac{dv}{dt} &= -e^T (x - (x + ve + w - ax + u)^+) + e^T x - k \\
\epsilon \frac{dw}{dt} &= -x + (x + ve + w - ax + u)^+ - y + (y + w)^+ + x + y - e
\end{align*}
\]

where $x, y, w \in \mathbb{R}^n$, $v \in \mathbb{R}$, $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$, $\epsilon > 0$, $x^+ = (x_1^+, \ldots, x_n^+)^T$, and $x_i^+ = \max\{0, x_i\}$. 
QP-based Projection Network

The projection neural network for $k$WTA operation based on the QP formulation needs $n + 1$ neurons and $2n + 2$ connections, which dynamic equations can be written as:

$$\begin{cases}
\epsilon \frac{dx}{dt} = -x + g(x - \eta(ax - ye - u)), \\
\epsilon \frac{dy}{dt} = -e^T x + k.
\end{cases}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $\epsilon$ and $\eta$ are positive constants, $g(x) = (g(x_1), \ldots, g(x_n))^T$ and

$$g(x_i) = \begin{cases} 
0, & \text{if } x_i < 0, \\
x_i, & \text{if } 0 \leq x_i \leq 1, \\
1, & \text{if } x_i > 1.
\end{cases}$$
LP-based Projection Network

Based on the equivalent LP formulation, we propose a recurrent neural network for KWTA operation with its dynamical equations as follows:

\[
\begin{align*}
\epsilon \frac{dx}{dt} &= -x + g(x + \alpha ey + \alpha u), \\
\epsilon \frac{dy}{dt} &= e^T x - k,
\end{align*}
\]

where \( \epsilon > 0, \alpha > 0, x \in R^n, y \in R. \)
QP-based Simplified Dual Net

The simplified dual neural network for $k$-WTA operation based on the QP formulation needs $n$ neurons and $3n$ connections, and its dynamic equation can be written as:

$$\begin{align*}
\epsilon \frac{dy}{dt} & = -M y + g((M - I)y - s) - s \\
x & = My + s,
\end{align*}$$

where $x, y \in \mathbb{R}^n$, $M = 2(I - ee^T/n)/a$, $s = Mu + ke/n$, $I$ is an identity matrix, $\epsilon$ and $g$ are defined as before.

---

LP-based One-layer $k$ WTA Net

The dynamic equation of a new LP-based $k$ WTA network model is described as follows:

$$\epsilon \frac{dx}{dt} = -Px - \sigma(I - P)g(x) + s,$$

where $P = ee^T/n$, $s = u - Pu + ke/n$, $\epsilon$ is a positive scaling constant, $\sigma$ is a nonnegative gain parameter, and $g(x) = (g(x_1), g(x_2), \ldots, g(x_n))^T$ is a discontinuous vector-valued activation function.
A discontinuous activation function is defined as follows:

\[
g(x_i) = \begin{cases} 
1, & \text{if } x_i > 1, \\
[0, 1], & \text{if } x_i = 1, \\
0, & \text{if } 0 < x_i < 1, \\
[-1, 0], & \text{if } x_i = 0, \\
-1, & \text{if } x_i < 0. 
\end{cases}
\]
Activation Function (cont’d)

\[ g(x_i) \]

Diagram:
- Vertical axis: \( g(x_i) \)
- Horizontal axis: \( x_i \)
- Points:
  - \( x_i = 0 \), \( g(0) = -1 \)
  - \( x_i = 1 \), \( g(1) = 1 \)
Convergence Results

The network can perform the $k$WTA operation if $\bar{\Omega} \subset \{x \in \mathbb{R}^n : 0 \leq x \leq 1\}$, where $\bar{\Omega}$ is the set of equilibrium point(s).

The network can perform the $k$WTA operation if it has a unique equilibrium point and $\sigma \geq 0$ when $(I - ee^T/n)u = 0$ or one of the following conditions holds when $(I - ee^T/n)u \neq 0$:

(i) $\sigma \geq \frac{\sum_{i=1}^{n} |u_i - \sum_{j=1}^{n} u_j/n|}{2n-2}$, or

(ii) $\sigma \geq n \sqrt{\frac{\sum_{i=1}^{n} (u_i - \sum_{j=1}^{n} u_j/n)^2}{n(n-1)}}$, or

(iii) $\sigma \geq 2 \max_i |u_i - \sum_{j=1}^{n} u_j/n|$, or,

(iv) $\sigma \geq \frac{\sqrt{\sum_{i=1}^{n} (u_i - \sum_{j=1}^{n} u_j/n)^2}}{\min_{\gamma_i \in \{-1,0,1\}}^+ \{ |\sum_{i=1}^{n} (u_i - \sum_{j=1}^{n} u_j/n)\gamma_i| \}}$. 
Simulation Results

Consider a $k$WTA problem with input vector $u_i = i \ (i = 1, 2, \ldots, n)$, $n = 5$, $k = 3$.

Transient behaviors of the $k$WTA network $\sigma = 6$. 
Transient behaviors of the $k$WTA network with $\sigma = 2$. 
Simulation Results (cont’d)

Convergence behavior of the $k$WTA network with respect to different values of $n$. 

Computational Intelligence Laboratory, CUHK – p. 58/69
QP-based One-layer $k$WTA Net

A QP-based $k$WTA network model with a discontinuous activation function is described as follows:

\[
\epsilon \frac{dz}{dt} = -(I - P)z - [aI + (1 - a)P]g(z) + s,
\]

\[
x = -\frac{1}{a}(I - P)z + \frac{s}{a} + \frac{k(a - 1)}{na}e,
\]

where \( g(z) = (g(z_1), g(z_2), \ldots, g(z_n))^T \) is a discontinuous activation function and \( \epsilon \) is a positive scaling constant.
Activation Function

\[ g(z_i) = \begin{cases} 
1, & \text{if } z_i > 0, \\
[0, 1], & \text{if } z_i = 0, \\
0, & \text{if } z_i < 0.
\end{cases} \]
Convergence Results

The neural network with any $a > 0$ is stable in the sense of Lyapunov and any trajectory is globally convergent to an equilibrium point.

$$x^* = -(I - P)z^*/a + s/a + (a - 1)ke/(na)$$ is an optimal solution of $k$WTA problem, where $z^*$ is an equilibrium point of the neural network.
Simulation Results
Simulation Results (cont’d)
Simulation Results (cont’d)

![Graph showing time (sec) versus z axis with curves for different values of α and n, including α = 0.01, n = 5; α = 0.10, n = 5; α = 0.50, n = 5; α = 1.00, n = 5. The graph includes labels for z sub 1 and z sub 5.]
Simulation Results (cont’d)

![Simulation Results Diagram]

- $n=5, \alpha=0.01$
- $n=10, \alpha=0.01$
- $n=15, \alpha=0.01$
- $n=20, \alpha=0.01$

Computational Intelligence Laboratory, CUHK – p. 65/69
A Dynamic Example

Let inputs be 4 sinusoidal input signals (i.e., $n = 4$)

$$u_i(t) = 10 \sin[2\pi(1000t + 0.2(i - 1))]$$

and $k = 2$. 
# Model Comparisons

<table>
<thead>
<tr>
<th>model</th>
<th>layer(s)</th>
<th>neurons</th>
<th>connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP-based primal-dual network</td>
<td>2</td>
<td>$n + 1$</td>
<td>$2n + 2$</td>
</tr>
<tr>
<td>QP-based primal-dual network</td>
<td>2</td>
<td>$3n + 1$</td>
<td>$6n + 2$</td>
</tr>
<tr>
<td>LP-based projection network</td>
<td>2</td>
<td>$n + 1$</td>
<td>$2n + 2$</td>
</tr>
<tr>
<td>QP-based projection network</td>
<td>2</td>
<td>$n + 1$</td>
<td>$2n + 2$</td>
</tr>
<tr>
<td>QP-based simplified dual network</td>
<td>1</td>
<td>$n$</td>
<td>$3n$</td>
</tr>
<tr>
<td>LP-based one-layer network</td>
<td>1</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
<tr>
<td>QP-based one-layer network</td>
<td>1</td>
<td>$n$</td>
<td>$3n$</td>
</tr>
</tbody>
</table>

---

Concluding Remarks

Neurodynamic optimization has been demonstrated to be a powerful alternative approach to many optimization problems.

For convex optimization, recurrent neural networks are available with global convergence to the optimal solution.

Neurodynamic optimization approaches provide parallel distributed computational models more suitable for real-time applications.
Future Works

The existing neurodynamic optimization model can still be improved to reduce their model complexity or increase their convergence rate.

The available neurodynamic optimization model can be applied to more areas such as control, robotics, and signal processing.

Neurodynamic approaches to global optimization and discrete optimization are much more interesting and challenging.

It is more needed to develop neurodynamic models for nonconvex optimization and combinatorial optimization.